Tableau Systems for the Modal μ-Calculus

Natthapong Jungteerapanich

Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2010

Abstract

The main content of this thesis concerns a tableau method for solving the satisfiability problem for the modal μ-calculus. A sound and complete tableau system for the modal μ-calculus is given. Since every tableau in such tableau system is finite and bounded by the length of the formula, the tableau system may be used as a decision procedure for determining the satisfiability of the formula. An alternative proof of the small model property is obtained: every satisfiable formula has a model of size single-exponential in the length of the formula. Contrary to known proofs in literature, the results presented here do not rely on automata theory.

Two simplifications of the tableau system are given. One is for the class of aconjunctive formulae. The resulting tableau system has been used to prove the completeness of Kozen's axiomatisation with respect to the aconjunctive fragment of the modal μ-calculus. Another is for the formulae in the class Π_2^μ.

In addition to the tableau method, the thesis explores some model-surgery techniques with the aim that such techniques may be used to directly prove the small model theorem. The techniques obtained so far have been used to show the small model property for Π_2^μ-formulae and for formulae with linear models.

Declaration

I declare that this thesis was composed by myself, that the work herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Natthapong Jungteerapanich

Acknowledgements

First and foremost, I owe my deepest gratitudes to my supervisor, Prof. Colin Stirling, for his guidance and support throughout my PhD study. It was from him that I learned the skills to tackle difficult research problems. His attention to detail and his high standard for clarity made me learn to think and write as clearly as possible. I truly appreciate his devotion in supervising me.

I would also like to thank Dr. Julian Bradfield, who was my second supervisor, Dr. Kousha Etessami, who was one of my PhD committees, for their comments in the progress meetings. Special thanks to Dr. Mark Jerrum for helping with some combinatorial problems related to my research. Further thanks to LFCS and the School of Informatics for providing an excellent research environment. Thanks to Andrew Finnie of the Informatics Graduate School for assisting me many times with administrative matters.

I also wish to thank Dr. Visit Hirankitti, who was my undergraduate supervisor at KMITL, Thailand, for introducing me to logic, and to Dr. Krysia Broda and Prof. Marek Sergot, who were my supervisors during my master-degree study at Imperial College London, for introducing me to automated reasoning and modal logic.

My PhD research was financially supported by the Oversea Research Studentship (ORS) award and University of Edinburgh scholarship. I must thank my sponsors for making my PhD study possible. I would also like to thank the School of Informatics and my parents for helping me with my living expense.

And for the last stage of the PhD process, I would like to thank to Dr. Richard Mayr and Dr. Yde Venema for examining this thesis and pointing out important errors and suggesting various improvements in the first version of the thesis. Any other errors remaining in this thesis are, of course, due to my own negligence.

Finally, I would like to thank my family and my colleagues for supporting me in every way. Particularly, I must thank my sister for keeping me well-fed during my later years in Edinburgh.

Contents

0 Preliminaries **1**

0.1 Numbers, Ordinals, Sets, Functions, and Relations 1
0.2 Words and Trees 1
0.3 Lattices 2
0.4 Fixpoint Theorems 3
0.5 Well Quasi-Orderings 5
0.6 Transition Systems 6
0.7 Automata on Infinite Words 7
0.8 Well-Known Theorems 8

1 Introduction **9**

1.1 Logic in Verification 9
1.2 Modal μ-Calculus 11
1.3 Goals 12
1.3.1 Satisfiability Problem 12
1.3.2 Small Model Property 13
1.4 Contributions 14
1.5 Outline 15

2 Modal μ-Calculus **16**

2.1 Syntax 16
2.2 Semantics 20
2.3 Basic Invariance Results 27
2.4 Alternation 29
2.5 Axiomatisation 31

3 Pre-Models and Tableaux **37**

3.1 Pre-Models 39
3.2 Signatures 43
3.3 Fundamental Semantic Theorem 47
3.3.1 Some Applications 49
3.4 Tableau Methods 52

3.4.1 Tableau Systems 53
3.4.2 Tableau System TS_0 54
3.4.3 Guardedness and Tableaux. 58

4 Tableau Systems for the Modal μ-Calculus **61**
4.1 Motivation 61
4.2 Tableau System ACON 65
4.2.1 Soundness 74
4.2.2 Completeness 77
4.2.3 Relation to Kozen's Tableau System 82
4.3 Tableau System TS 82
4.3.1 Soundness 95
4.3.2 Completeness 102
4.3.3 Small Model Property 108
4.3.4 Complexity 108
4.3.5 Relation to Safra Construction 110
4.4 Axiomatic Completeness 114
4.4.1 Canonical Models 114
4.4.2 Completeness for Aconjunctive Fragment 115

5 Model Surgery **123**
5.1 Operations on Models and Pre-Models 124
5.2 Model Surgery Using Trails 126
5.2.1 Trail Equivalence 126
5.2.2 Safe Pairs of States 131
5.2.3 Small Model Theorem: Linear Case 133
5.3 Π_2^μ-Formulae 134
5.3.1 Small Model Theorem for Π_2^μ 135
5.3.2 Tableau System NUMU 140

6 Conclusion **149**

List of Figures

2.1 Model $\mathcal{M}$ in Example 2.10. 23
2.2 Model $\mathcal{M}$ in Example 2.11. 23

4.1 A fragment of a TS_0-tableau for ϕ in Example 4.1 63
4.2 Some fragments of TS_0-tableaux for Example 4.2. 64
4.3 A successful ACON-tableau for the formula ϕ in Example 4.1 69
4.4 An unsuccessful ACON-tableau for Example 4.6 70
4.5 An unsuccessful ACON-tableau for the satisfiable formula ϕ in Example 4.7 71
4.6 An unsuccessful TS-tableau for Example 4.33 87
4.7 Another unsuccessful TS-tableau for Example 4.33 87
4.8 A successful TS-tableau for Example 4.34 88
4.9 A successful TS-tableau for Example 4.35 89
4.10 TS-Tableaux for Example 4.37 91
4.11 A TS-Tableau for Example 4.38 92

5.1 Jumping on a tree transition system 125
5.2 Pre-model $\mathcal{P}$ in Example 5.12 130
5.3 Pre-model $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_n,s_m}(\mathcal{P}))$ 134
5.4 Pre-model $\mathcal{P}''$ 134
5.5 Joining small acyclic pre-models 139
5.6 A successful NUMU-tableau for Example 5.30. 142
5.7 A tree of unsuccessful NUMU-tableaux for Example 5.31. 143
5.8 An unsuccessful NUMU-tableau for Example 5.32. 144

Chapter 0

Preliminaries

0.1 Numbers, Ordinals, Sets, Functions, and Relations

$\mathbb{N} = \{0, 1, 2, ...\}$ denotes the set of *natural numbers*. $\mathbb{O}$ denotes the class of *ordinals*. We assume the standard well ordering $\leq$ (and $<$) on $\mathbb{N}$ and $\mathbb{O}$. ω denotes the initial limit ordinal. The *cardinality of set* A is denoted by $|A|$. The powerset of A is denoted by $\wp(A)$.

The *domain* of a function f is denoted by $\mathrm{Dom}(f)$. Given a function f and a subset A of $\mathrm{Dom}(f)$, we use $f(A)$ to denote the set $\{f(a) \mid a \in A\}$; the *restriction of* f *to* A, written $f \restriction A$, is the function with domain A such that $(f \restriction A)(a) = f(a)$ for each $a \in A$. For any element a (possibly in $\mathrm{Dom}(f)$), we use $f[a := b]$ to denote the function with domain $\mathrm{Dom}(f) \cup \{a\}$ such that $f[a := b](a) = b$ and $f[a := b](a') = f(a')$ for each $a' \in \mathrm{Dom}(f)$, $a' \neq a$.

Unless stated otherwise, by a *relation*, we mean a *binary* relation $R \subseteq A \times A$ on some set A. Given a relation $R \subseteq A \times A$, a *path over* R is a (finite or infinite) sequence of pairs of the form $(a_1, a_2), (a_2, a_3), ...$, where each $(a_i, a_{i+1}) \in R$. A finite path $(a_1, a_2), (a_2, a_3), ..., (a_{n-1}, a_n)$, $n \geq 2$ can thus be written compactly as $a_1 R a_2 R ... R a_n$; and similarly an infinite path $(a_1, a_2), (a_2, a_3), ...$ as $a_1 R a_2 R ....$ The *length* of a finite path is the number of pairs in it.

0.2 Words and Trees

Let A be a non-empty set, called an *alphabet*.

A *finite word* w over A of length $n \in \mathbb{N}$ is a function from $\{0, 1, ..., n-1\}$ to A, usually written as $w(0)w(1)w(2)...w(n-1)$. $|w|$ denotes the length of w. The unique word of length 0 is denoted by ϵ. A^* denotes the set of all finite words over A. An *ω-word* over A is a function from $\mathbb{N}$ to A. A^ω denotes the set of all ω-words over A.

Given a finite word u and a (finite or ω-) word v over the same alphabet, the *concatenation* of u and v, written uv, is the word w where $w(i) = u(i)$ for each $i < |u|$ and $w(|u| + j) = v(j)$ for each $j \geq 0$. u is said to be a *prefix of* v iff $\mathrm{Dom}(u) \subseteq \mathrm{Dom}(v)$

and $u(i) = v(i)$ for each $i < |u|$; u is said to be a *proper prefix* of v iff u is a prefix of v and $u \neq v$.

A *tree* T over a label set L is a function from $\mathrm{Dom}(T)$ to L where

- $\mathrm{Dom}(T)$ is a set of finite words (over some alphabet A) and
- $\mathrm{Dom}(T)$ is closed under prefixes (i.e. if u is in $\mathrm{Dom}(T)$ then so is every prefix of u).

Each element in $\mathrm{Dom}(T)$ is called a *node* in T. Since $\mathrm{Dom}(T)$ is closed under prefixes, T contains node ϵ, called the *root* of T. For each node u of T, $T(u)$ is called the *label* of u in T.

Suppose u and v are nodes in T. u is said to be a *parent* of v or, equivalently, v a *child* of u, iff $v = ua$ for some letter $a \in A$. u is said to be an *ancestor* of v or, equivalently, v a *descendant* of u iff u is a prefix of v (hence each node is both an ancestor and a descendant of itself). u is a *proper ancestor* of v or, equivalently, v a *proper descendant* of u iff u is a proper prefix of v. A *leaf* is a node with no children.

A *branch* in T is a maximal sequence $u_0, ..., u_n, ...$ of nodes in T such that u_0 is the root and each u_{i+1} is a child of u_i (thus a branch is either an infinite sequence or a finite sequence ending with a leaf).

The *degree of node* u is the cardinality of the set of its children. The *degree of tree* T is the least upper bound of the degrees of its nodes. A tree T is said to be a *finite tree* iff $\mathrm{Dom}(T)$ is finite.

0.3 Lattices

We recap some definitions from lattice theory ([DP90], [Bir93]).

A *partial ordering* $\sqsubseteq$ on a set A is a relation on A satisfying the following:

- **Reflexivity.** $a \sqsubseteq a$ for all $a \in A$;
- **Transitivity.** $a \sqsubseteq b$ and $b \sqsubseteq c$ implies $a \sqsubseteq c$ for all $a, b, c \in A$;
- **Antisymmetry.** $a \sqsubseteq b$ and $b \sqsubseteq a$ implies $a = b$ for all $a, b \in A$.

A *partially-ordered set* (or *poset*) is a pair $\langle A, \sqsubseteq \rangle$ where $\sqsubseteq$ is a partial ordering on A. "$a \sqsubseteq b$" is usually read "a is less than or equal to b" or "b is greater than or equal to a".

Suppose $\langle A, \sqsubseteq \rangle$ is a poset. The *least (greatest) element* of a set $B \subseteq A$, if exists, is the unique element $b \in B$ such that $b \sqsubseteq a$ (respectively, $a \sqsubseteq b$) for each $a \in A$. A *minimal (maximal) element* of $B \subseteq A$ is an element $b \in B$ such that, for each $a \in A$, $a \sqsubseteq b$ (respectively, $b \sqsubseteq a$) implies $a = b$. An *upper bound* of $B \subseteq A$ is an element $a \in A$ such that $b \sqsubseteq a$ for each $b \in B$. The *least upper bound (l.u.b.)* of B, if exists, is the least of all upper bounds of B. Dually, a *lower bound* of $B \subseteq A$ is an element $a \in A$ such that $a \sqsubseteq b$ for each $b \in B$. The *greatest lower bound (g.l.b.)* of B, if exists, is the greatest of all lower bounds of B. The l.u.b and the g.l.b. of a set $B \subseteq A$ are usually denoted by $\bigsqcup B$ and $\bigsqcap B$, respectively.

A *lattice* is a poset $\langle A, \sqsubseteq \rangle$ in which the l.u.b. and the g.l.b. of each pair $\{a, b\}$ of elements exist. A lattice is said to be *complete* iff the l.u.b. and the g.l.b. of any set $B \subseteq A$ exist. Thus every complete lattice has the least element, usually denoted by $\bot$, and the greatest element, usually denoted by $\top$, which are the g.l.b. and the l.u.b. of A, respectively. The lattices which are used extensively are the *(complete) powerset lattices* $\langle \wp(S), \subseteq \rangle$. It is clear that these lattices are complete (for any $A \subseteq \wp(S)$, the l.u.b. and the g.l.b. of A are $\bigcup A$ and $\bigcap A$, respectively).

0.4 Fixpoint Theorems

Let $\langle A, \sqsubseteq \rangle$ be a poset. A function $f : A \to A$ is said to be *monotone* iff $a \sqsubseteq b$ implies $f(a) \sqsubseteq f(b)$ for all $a, b \in A$.

An element $a \in A$ is a *pre-fixpoint of* f iff $f(a) \sqsubseteq a$, a *post-fixpoint of* f iff $a \sqsubseteq f(a)$, and a *fixpoint of* f iff $f(a) = a$. The *least (greatest) fixpoint of* f is the least (greatest) of all fixpoints of f; and similarly for the least or greatest pre-fixpoint and post-fixpoint. The least fixpoint of f, if exists, is denoted by μf and the greatest fixpoint of f by νf.

Theorem 0.1 (Knaster-Tarski Theorem [Tar55])**.** *Let $\langle A, \sqsubseteq \rangle$ be a complete lattice. Every monotone function $f : A \to A$ has the least fixpoint μf, which is the g.l.b. of all pre-fixpoints, and the greatest fixpoint νf, which is the l.u.b. of all post-fixpoints of f, i.e.*

$$\begin{aligned} \mu f &= \bigsqcap \{a \in A \mid f(a) \sqsubseteq a\} \\ \nu f &= \bigsqcup \{a \in A \mid a \sqsubseteq f(a)\}. \end{aligned}$$

Proof. Let a_* denote $\bigsqcap \{a \in A \mid f(a) \sqsubseteq a\}$. For each pre-fixpoint a, we have $a_* \sqsubseteq a$ which by monotonicity implies that $f(a_*) \sqsubseteq f(a) \sqsubseteq a$. Hence $f(a_*)$ is a lower bound of $\{a \in A \mid f(a) \sqsubseteq a\}$ and thus $f(a_*) \sqsubseteq a_*$. This means that a_* is the least pre-fixpoint. By monotonicity, $f(f(a_*)) \sqsubseteq f(a_*)$, i.e. a pre-fixpoint, which implies that $a_* \sqsubseteq f(a_*)$. Thus a_* is the least fixpoint μf of f.

The greatest-fixpoint case can be shown similarly. □

Approximants. The least fixpoint and the greatest fixpoint of a monotone function can be found in a more constructive way. For any monotone function $f : A \to A$, ordinal α and limit ordinal λ, define

$$\begin{aligned} f^0(\bot) &= \bot \\ f^{\alpha+1}(\bot) &= f(f^\alpha(\bot)) \\ f^\lambda(\bot) &= \bigsqcup_{\alpha<\lambda} f^\alpha(\bot). \end{aligned}$$

and

$$\begin{aligned} f^0(\top) &= \top \\ f^{\alpha+1}(\top) &= f(f^\alpha(\top)) \\ f^\lambda(\top) &= \bigsqcap_{\alpha<\lambda} f^\alpha(\top). \end{aligned}$$

The sequence $f^0(\bot), f^1(\bot), ...$ is monotone (i.e. $f^\alpha(\bot) \sqsubseteq f^{\alpha'}(\bot)$ if $\alpha \leq \alpha'$) and converges to the least fixpoint of f. If the lattice is finite, there must be some $n < \omega$ such that $f^n(\bot) = f^{n+1}(\bot)$ which must be equal to μf; hence this iterative process yields an algorithm for constructing the least fixpoint of f. For infinite complete lattices, we may need a transfinite number of iterations to reach a point where $f^\alpha(\bot) = f^{\alpha+1}(\bot)$ for some ordinal α. To construct the greatest fixpoint of f, we start with $f^0(\top)$ instead. The sequence $f^0(\top), f^1(\top), ...$ is anti-monotone (i.e. $f^{\alpha'}(\top) \sqsubseteq f^\alpha(\top)$ if $\alpha \leq \alpha'$) and converges to the greatest fixpoint of f. For this reason, $f^\alpha(\bot)$ and $f^\alpha(\top)$ are called the *approximants* for μf and νf, respectively.

The bound on the number of iterations required can be given by the height of the lattice. The *height* of a lattice is defined to be the least ordinal β such that every monotone sequence of *distinct* elements in the lattice has length no greater than β (we know from set theory that there must be such an ordinal β for every lattice). The height of a powerset lattice $\langle \wp(S), \subseteq \rangle$ is thus the least ordinal of cardinality *greater than* $|S|$. We thus have the following result.

Theorem 0.2. *Let f be a monotone function on a complete lattice $\langle A, \sqsubseteq \rangle$ of height β.*

$$\begin{aligned} \mu f &= \bigsqcup_{\alpha<\beta} f^\alpha(\bot) = f^\beta(\bot) \\ \nu f &= \bigsqcap_{\alpha<\beta} f^\alpha(\top) = f^\beta(\top) \end{aligned}$$

Proof. It is easy to show (using transfinite induction) that $f^\alpha(\bot) \leq \mu f$, for each α, and that the sequence $\langle f^\alpha(\bot) \rangle_{\alpha \leq \beta}$ is monotone. Since the lattice is of height β, there must be some $\gamma < \beta$ such that $f^\gamma(\bot) = f^{\gamma+1}(\bot) = ... = f^\beta(\bot)$. This implies that $\bigsqcup_{\alpha<\beta} f^\alpha(\bot) = f^\gamma(\bot)$ is a fixpoint. Since $f^\gamma(\bot) \leq \mu f$, it follows that $\bigsqcup_{\alpha<\beta} f^\alpha(\bot) = f^\beta(\bot)$ is the least fixpoint of f.

The greatest fixpoint case can be shown similarly. □

A monotone function $f : A \to A$ is said to be $\bigsqcup$-*continuous* (or, simply, *continuous*) iff, for any non-decreasing sequence $a_1 \sqsubseteq a_2 \sqsubseteq ....$ of elements in A,

$$\bigsqcup_{i<\omega} f(a_i) = f(\bigsqcup_{i<\omega} a_i).$$

Similarly, a monotone function f is said to be $\bigsqcap$-continuous iff, for any non-increasing

sequence $a_1 \sqsupseteq a_2 \sqsupseteq ...$ of elements in A,

$$\bigsqcap_{i<\omega} f(a_i) \quad = \quad f(\bigsqcap_{i<\omega} a_i).$$

The least fixpoint of a ($\bigsqcup$-) continuous function f can be approximated by finite approximants $f^i(\bot)$, $i < \omega$; namely, $\mu f = \bigsqcup_{i<\omega} f^i(\bot)$. Similarly, if f is $\bigsqcap$-continuous, then $\nu f = \bigsqcap_{i<\omega} f^i(\top)$.

0.5 Well Quasi-Orderings

In addition to partial orderings, we employ some basic result from the theory of well quasi-ordering. For more extensive treatment of the subject, we refer to [Kru54], [Kru60], [Kru72], and [Lav76].

A *quasi-ordering* $\sqsubseteq$ on a set A is any reflexive and transitive relation on A. Thus every partial ordering is a quasi-ordering. A *quasi-ordered set* (or *qoset*) is a pair $\langle A, \sqsubseteq\rangle$ where $\sqsubseteq$ is a quasi-ordering on A. Most terminology on partial orderings can be given for quasi-orderings in the same way.

Definition 0.3 (Well Quasi-Orderings)**.** A *well quasi-ordered set* is a qoset $\langle A, \sqsubseteq\rangle$ which satisfies any of the following equivalent conditions.

(1) $\sqsubseteq$ is a *well-founded* relation (i.e. there is no infinite strictly-descending chain, $... \sqsubset a_1 \sqsubset a_0$), and there is no infinite set of $\sqsubseteq$-incomparable elements.
(2) Every subset of A has a *finite base*: for all $B \subseteq A$, there exists a finite $B_0 \subseteq B$ such that for each $b \in B$ there is an $a \in B_0$ such that $a \sqsubseteq b$.
(3) Every countable sequence $a_0, a_1, ...$ contains a countable monotone subsequence $a_{i_0} \sqsubseteq a_{i_1} \sqsubseteq ...$ ($i_j < i_{j+1}$ for all $j \geq 0$)

The proof of the equivalence of the above conditions can be found in the mentioned reference.

By definition, every *well-ordered set* is a well quasi-ordered set. What we are interested in are the constructions which produce a new well quasi-ordered set from known ones. Here are some basic constructions.

The *disjoint union* of a family $\langle A_i, \sqsubseteq_i\rangle$, $i \in I$, of *disjoint* qosets is $\langle \bigcup_{i\in I} A_i, \bigcup_{i\in I} \sqsubseteq_i\rangle$.

The *product* of a family $\langle A_i, \sqsubseteq_i\rangle$, $i \in I$, of qosets is $\langle A^I, \sqsubseteq\rangle$ where A^I is the set of all functions $f : I \to A$ and $\sqsubseteq$ is defined component-wise: $f \sqsubseteq g$ iff $f(i) \sqsubseteq g(i)$ for each $i \in I$.

A *homomorphism* from a qoset $\langle A, \sqsubseteq\rangle$ into a qoset $\langle A', \sqsubseteq'\rangle$ is a function $h : A \to A'$ such that $a \sqsubseteq b$ implies $h(a) \sqsubseteq' h(b)$ for each $a, b \in A$. For any homomorphism $h : A \to A'$, $\langle h(A), \sqsubseteq''\rangle$, where $\sqsubseteq''$ is the restriction of $\sqsubseteq'$ to $h(A)$, is called a *homomorphic image* of $\langle A, \sqsubseteq\rangle$.

Proposition 0.4.

(a) Any subset of a well-quasi ordered set is a well-quasi ordered set.

(b) The disjoint union of a finite *family of* disjoint *well-quasi ordered sets is a well-quasi ordered set.*

(c) The product of a finite *family of well-quasi ordered sets is a well-quasi ordered set.*

(d) Any homomorphic image of a well-quasi ordered set is a well-quasi ordered set.

Proof. The proof is not difficult and can be found in the mentioned references. □

0.6 Transition Systems

A *transition system* (also called a *labelled transition system*) is a pair $\mathcal{S} = \langle S, \{R_a\}_{a\in A}\rangle$, where

- S is a set of *states*,
- A is a non-empty set of *labels*,
- $R_a \subseteq S \times S$, for each $a \in A$, is a binary relation on S.

It is sometimes convenient to look at each relation R_a and its inverse R_a^{-1} as functions over sets of states where, for any set S' of states:

$$\begin{aligned} R_a(S') &= \{t \mid sR_at, s \in S'\} \\ R_a^{-1}(S') &= \{s \mid sR_at, t \in S'\}. \end{aligned}$$

We shall use the functional and relational representations of R_a interchangeably.

The *degree of a state s in $\mathcal{S}$* is the cardinality of $\bigcup_{a\in A} R_a(\{s\})$. The *degree of a transition system* is the l.u.b. of the degrees of its states.

A *path in a transition system $\mathcal{S}$* is a (finite or ω-) word over triples (s, a, t), where $a \in A$ and sR_at, in which, for each pair of consecutive elements $(s_i, a_i, t_i)(s_{i+1}, a_{i+1}, t_{i+1})$, $t_i = s_{i+1}$; thus a path can be written compactly as: $s_0 \to_{a_0} s_1 \to_{a_1} \ldots \to_{a_n} s_{n+1} \to_{a_{n+1}} \ldots$ A *path from state s* is one where the first state is s. A (finite) *path to state t* is one where the last state is t. A *cycle* is a finite path from some state s to s itself.

We say that a transition system $\mathcal{S}' = \langle S', \{R'_a\}_{a\in A}\rangle$ is *contained in* another transition system $\mathcal{S} = \langle S, \{R_a\}_{a\in A}\rangle$ (with the same label set A), or that $\mathcal{S}'$ is a *subsystem of* $\mathcal{S}$, iff $S' \subseteq S$ and each R'_a is the restriction R_a to domain S'. Clearly, for any subset $T \subseteq S$, there is a unique subsystem of $\mathcal{S}$ whose states are exactly T.

Definition 0.5. Roughly speaking, a *tree transition system* is a transition system which is isomorphic to a tree. Precisely, a *tree transition system* is a transition system $\mathcal{S} = \langle S, \{R_a\}_{a\in A}\rangle$ for which there exists a tree T labelled by S such that

- T is a one-one correspondence from $\mathrm{Dom}(T)$ onto S and
- for each node u and v, v is a child of u iff, for some $a \in A$, $T(u)R_aT(v)$.

It is easy to see that there must be a state s_0 such that, for any tree T satisfying the above conditions, $T(\epsilon) = s_0$. s_0 is thus called the *root* of $\mathcal{S}$. The standard terminology for trees can be given for tree transition systems in the obvious way: t is a *child* of s iff sR_at for some action a; t is a *descendant* of s iff either $s = t$ or there is a path from s to t. Other notions, such as leaf, parent, ancestor etc., can be given similarly.

Definition 0.6. Given a tree transition system $\mathcal{S}$ and a state s, by a *partial subtree of $\mathcal{S}$ rooted at s*, we mean a subsystem $\mathcal{S}'$ of $\mathcal{S}$ which satisfies the following:

- $\mathcal{S}'$ is a tree transition system with root s; and
- for each state s' in $\mathcal{S}'$, if s' has a child in $\mathcal{S}'$, then all children of s' in $\mathcal{S}$ are included in $\mathcal{S}'$.

Obviously, there can be many partial subtrees of $\mathcal{S}$ rooted at s; we call the largest one (which contains all descendants of s) the (complete) subtree of $\mathcal{S}$ rooted at s.

0.7 Automata on Infinite Words

For later discussion, we include here some terminology and definitions from the theory of automata on infinite words. More extensive treatment of the subject can be found in many places, including [Tho90],[Wal01],[VW94].

A *(nondeterministic) automaton on infinite words* is given by $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \text{Acc} \rangle$ where Σ is an alphabet, Q is a finite set of states, $q_0 \in Q$ (called the *initial state*), $\delta : Q \times \Sigma \to \wp Q$ (called the *transition function*), and Acc is the *acceptance condition* (see below). An automaton $\mathcal{A}$ is said to be *deterministic* iff, for each $q \in Q$ and $a \in \Sigma$, $|\delta(q, a)| = 1$.

A *run* of $\mathcal{A}$ on a word $w = a_0a_1a_2 \in \Sigma^\omega$ is a word $r = q_0q_1q_2... \in Q^\omega$ such that q_0 is the initial state and $q_{i+1} \in \delta(q_i, a_i)$ for each $i \geq 0$. A run is *accepting* iff it *satisfies* the acceptance condition Acc. This is defined for each type of acceptance conditions as follows:

- **Büchi condition**: Acc is given by a set $F \subseteq Q$. A run r satisfies the Büchi condition F iff some state in F occurs infinitely often on r.
- **Rabin condition**: Acc is given by a set of pairs $C = \{(R_1, G_1), ..., (R_n, G_n)\}$ where $R_i, G_i \subseteq Q$. A run r satisfies such Rabin condition C iff there is some i such that each state in R_i occurs *finitely* often on r and some state in G_i occurs *infinitely* often on r.
- **Streett condition**: Acc is given by a set of pairs $C = \{(R_1, G_1), ..., (R_n, G_n)\}$ where $R_i, G_i \subseteq Q$. A run r satisfies such Streett condition C iff, for each i, if some state in R_i occurs *infinitely* often on r, then some state in G_i also occurs *infinitely* often on r.
- **Muller condition**: Acc is given by a set C of subsets of Q. A run r satisfies the Muller condition C iff there is a set $F \in C$ which contains precisely all the states occurring *infinitely* often on r.

- **Parity condition**: Acc is given by a function $\Omega : Q \to \mathbb{N}$. A run r satisfies the parity condition Ω iff the least number in $\{\Omega(q) \mid q \text{ occurs infinitely often on } r\}$ is *even*.

$\mathcal{A}$ accepts word $w \in \Sigma^\omega$ iff there is an accepting run of $\mathcal{A}$ on w. The *language recognised by* $\mathcal{A}$, denoted $\mathcal{L}(A)$, is the set of all words accepted by $\mathcal{A}$. Two automata are said to be *equivalent* iff they accept the same language.

By a *Büchi* (resp., *Rabin*, *Streett*, *Muller*, *parity*) automaton, we mean an automaton on infinite words with a *Büchi* (resp., *Rabin*, *Streett*, *Muller*, *parity*) condition.

0.8 Well-Known Theorems

Lemma 0.7 (König's Lemma [Kön27])**.** *Every infinite tree of finite degree contains an infinite branch.*

Proof. The proof can be found in many places, including [Smu68]. □

For any set A, let $A^{(m)}$ denote the set of all subsets of size m of A.

Theorem 0.8 (Ramsey's Theorem [Ram28])**.** *For any number m and any finite* partition $C_1 \cup ... \cup C_n = \mathbb{N}^{(m)}$*, there exists an infinite set $S \subseteq \mathbb{N}$ such that $S^{(m)} \subseteq C_i$ for some i.*

Proof. This was originally shown (in a more general form) in [Ram28]. □

Chapter 1

Introduction

1.1 Logic in Verification

One application of logic in computer science is program verification. Logic has long been used as a formal language for expressing specifications of programs with the aim that program correctness can be proved formally and mechanically. Early research focused on the input/output correctness of programs. A prominent example is the works by Floyd [Flo67] and Hoare [Hoa69]. *Floyd-Hoare logic* enables one to write a specification for a simple *while* program in the form of an assertion which specifies the precondition and the postcondition of the program. A formal proof system built upon a deductive system of the underlying logic is then used to prove the correctness of such assertions. Floyd-Hoare logic has since been expanded in many ways. An important development was by Pratt [Pra76], who suggested that Floyd-Hoare logic can be reformulated as a (first-order) modal logic. The logic, known as *Dynamic Logic*, views a program as a modality. Formulae in such logic can then be interpreted using the standard possible-world semantics. For example, an assertion in Floyd-Hoare logic can be written in the form $\phi \rightarrow [p]\psi$, which means that in every state where ϕ (the precondition) holds, ψ (the postcondition) holds at every state reachable via program p. The modal interpretation of programs provides a new understanding of program properties – the truth of a formula is *dynamic*, i.e. changing over the course of program executions. Dynamic logic, and particularly its propositional fragment *PDL* [FL79], has since been extensively studied not only in the computer science community but also in the modal logic community.

Another early use of modal logic is in the research on process calculi by Hennessy and Milner ([Mil80], [HM80], [HM85]). In this area, one studies the behaviour of processes, typically modelled by *labelled transition systems*, in response to the environment. An important question is when two processes are considered equivalent. This has led to the notion of *observational equivalence* [HM80] and *bisimulation* ([vBe76], [Par81], [HM80]). A simple (propositional) modal logic with a modality for each action, called *Hennessy-Milner logic* (HML), was introduced as another characterisation of equivalence. The

key result is that two image-finite processes are bisimulation equivalent if and only if they satisfy the same formulae in HML. From the verification point of view, Hennessey-Milner logic can be seen as a language for specifying properties of processes. However, being a simple modal logic, HML is quite weak in this aspect as only the properties involving a finite future of the process can be expressed.

This latter use of logic to specify a property on the transitional behaviour of processes is an example of the verification of *reactive systems*. In contrast to a program whose purpose is to compute an output from an input, a reactive system is a computing model which persistently interacts with the environment and changes its state accordingly. Examples of such systems abound, including operating systems, communication protocols, and various industrial control systems. Clearly, earlier verification techniques for input/output correctness do not apply. This has led to a new area of research in verification. A significant advance was made when Pnueli [Pnu77] proposed the use of *temporal logic* as a formalism for verifying reactive systems. The temporal logic considered in [Pnu77] contains two temporal operators, G and F, and the formulae are interpreted over a *run* in the transition system. Intuitively, a formula $\mathsf{G}\phi$ is true on a run in which ϕ holds at every state, $\mathsf{F}\phi$ on a run in which ϕ holds at some state, and a formula is true at a state if and only if it is true on every run from that state. Thus G is useful for expressing *safety* properties and F, on the other hand, is useful for expressing *liveness* properties.

Numerous temporal logics have since been proposed and studied. The mentioned logic by Pnueli was later developed into the *Linear-time Temporal Logic (LTL)*. For nondeterministic systems, a class of *branching-time* temporal logics where formulae are interpreted over all possible runs from the given state was also proposed. Prominent examples are the *Computation Tree Logic (CTL)* [CE81] and its extension CTL^* [EH86]. These logics allow one to specify properties which stipulate the existence (or non-existence) of certain runs. For example, a formula $\mathsf{EG}\phi$ in CTL states that there is a run where ϕ holds throughout. In the temporal-logic framework, important problems of logical and practical interest include

- **Model checking**: Find an algorithm which determines whether a formula is true in the given model. This is the most important task if the temporal logic is to be used for verification. The existence of a fast algorithm dictates whether the logic can be used in practice.
- **Expressiveness**: What properties can be expressed in the logic? Naturally one wants a logic which is expressive enough to represent all the required properties. However, more expressive logics usually come at the cost of higher computational complexity. More importantly, since the complexity generally depends on the length of the formula, one wants a logic which can *succinctly* represent the properties.
- **Satisfiability**: Find an algorithm which determines whether there is a model

which satisfies the given formula. Apart from its logical importance, there is research in program synthesis which uses such an algorithm to construct a model from which a program can be extracted.

- **Deductive completeness**: Is there a sound and complete deductive system for the logic? This question is of logical interest. Such a deductive proof system allows one to derive valid formulae in the logic.

1.2 Modal μ-Calculus

As a logic for specification, the modal μ-calculus is one of the most extensively studied. Essentially, it is an extension of a modal logic of actions (as in Hennessy-Milner logic [HM80]) with the least and greatest fixpoint operators. It was introduced by Kozen [Koz83] as an improvement of the logic of least roots proposed by Pratt [Pra81]. The idea of using fixpoint operators to extend the expressive power of the logic can, however, be traced back to De Bakker, De Roever, Scott and Park [Par69], among others. The important features of the modal μ-calculus that make it a very interesting formalism can be summarised below.

- **Expressiveness.** Modal μ-calculus strictly subsumes many well-known temporal and program logics. PDL ([Pra76], [FL79], [HKT00]) and PDLΔ [Str81] are some examples. In fact, one reason for the introduction of the modal μ-calculus is to make these logics more expressive. Both temporal logics CTL [CE81] and CTL* [EH86] have been shown to be less expressive than the modal μ-calculus ([Dam94] and [BC96]).

  What makes modal μ-calculus very expressive is the alternation of the least and greatest fixpoint operators in formulae. While simple properties, including liveness and termination (and those expressible in PDL, PDLΔ or CTL), require no alternation, more complex properties do so. In fact, it has been shown that allowing more alternation makes the logic more expressive ([Bra97], [Bra98], [Len96]).

  [JW96] proves that the modal μ-calculus is equi-expressive to the bisimulation-invariant fragment of the monadic-second order logic over graphs. This result is analogous to the van Benthem Characterisation Theorem in modal logic [vBe76].

- **Decidability and complexity.** Despite its expressiveness, modal μ-calculus is decidable and efficient to compute. Model checking in modal μ-calculus is an active area of research. The problem can be reduced to solving parity games over finite graphs, and has been shown to be in UP $\cap$ coUP [Jur98]. It is a famous open problem whether this can be reduced to polynomial time. Satisfiability checking is EXPTIME-complete (the upper bound follows from the results in [SE89], [EJ88], [Saf88]; the lower bound follows from the EXPTIME-completeness of PDL [FL79]). The algorithms for these tasks are the results of the fruitful connection with automata and game theory.

- **Automata.** The application of automata theory in temporal logics has long been studied ([VW86], [CES86], [VW94] etc.) and is arguably the main reason of their success. The bridge between the modal μ-calculus and the theory of automata on infinite objects was shown by Streett and Emerson in the landmark paper [SE89]. Since then automata have become invaluable tools for understanding and solving problems in the modal μ-calculus. The equivalence between the modal μ-calculus and the alternating parity automata has helped in proving important properties, the expressiveness result in [JW96] being one example. With this close connection, progress in one of these fields usually has applications in the other.

- **Axiomatisation.** The logic has a finite axiomatic system given in the original paper by Kozen [Koz83]. Kozen's axiomatisation was first proved sound and complete in [Wal00]. Other sound and complete axiomatisations are given in [Koz86], [Wal93], [AKM95], [BK95].

1.3 Goals

Our research has two main objectives. First, we wish to obtain a tableau system for the satisfiability problem of the modal μ-calculus which serves one or more of the following purposes:

(1) A decision procedure employing the tableau system to check the satisfiability of a formula can be written.
(2) The tableau system can be used as a tool for proving the completeness of Kozen's axiomatisation.
(3) The tableau system acts as an alternative characterisation which is useful for proving logical properties of the logic, including the small model property.

Secondly, we look for a direct proof of the small model theorem by means of model surgery. The idea is to study model-theoretic operations which can be used to transform arbitrary models into a small model for the given formula. We hope that by performing the studies we will come up with a new set of tools for proving properties of the logic.

In practice, we have found these two goals interrelated. Studying operations on models helps us design a tableau system. Conversely, the soundness and completeness proofs of tableau systems might shed some light on useful operations on models. We now turn to explain our motivation for undertaking these goals and the known results in literature.

1.3.1 Satisfiability Problem

The first known decision procedure for satisfiability of the modal μ-calculus was obtained using the reduction to the monadic second-order theory of n-successors (SnS) [KP84]. It is widely known that the modal μ-calculus can be embedded in SnS. The

satisfiability of SnS has long been known to be decidable as was first shown by Rabin [Rab69]. But this method is highly inefficient since the optimal decision procedure for SnS is known to be non-elementary.

A major milestone was made by Streett and Emerson [SE89], who introduced the notion of *well-founded pre-models* as a characterisation of models. The paper also suggested that the existence of a well-founded pre-model for the given formula can be checked by automata. Particularly, to show whether a formula is satisfiable, an infinite-tree automaton which accepts all well-founded tree pre-models for the formula is constructed; the formula is satisfiable iff the automaton accepts some tree (which can be seen as a tree model for the formula). This established a connection between the modal μ-calculus and automata on infinite objects. Since then, progress in the related automata theory has usually led to the improvement of the decision procedures (e.g. for satisfiability checking or model checking) for the modal μ-calculus. For example, an efficient determinisation construction on ω-word automata by Safra [Saf88] has led to an optimal decision procedure for satisfiability (determinisation or complementation of automata is the key element in the construction of the automaton recognising the well-founded tree pre-models of a formula). The use of automata together with Safra's construction has long been the only optimal solution to the satisfiability problem.

The automata-theoretic approach is not without disadvantages. Essential details of the procedure are hidden in the construction of automata and the emptiness-checking algorithm. This means that in order to prove properties of the logic we need to resort to work with those automata for formulae. This is why we aim to find a tableau system for satisfiability which is useful for proving properties of the logic. For example, if we could find a tableau system in which a successful tableau for a formula can be seen as a *finite* model for the formula, the soundness and completeness of such tableau system immediately implies the finite model property. Of particular interest is the use of tableaux as a tool for proving the completeness of an axiomatisation of the logic. This completeness-via-tableaux approach has been applied to many other logics, including first-order logic, various modal logics, LTL, CTL, and PDL. For the modal μ-calculus, a natural axiomatisation by Kozen [Koz83] has been shown to be sound and complete in [Wal00]. The proof is however highly intricate. We believe that if we could come up with a tableau system which exposes the right structure of a formula, the completeness of Kozen's axiomatisation can be shown using such tableau system.

1.3.2 Small Model Property

A logic is said to have the finite model property if every satisfiable formula is satisfied by a finite model. The small model property is stronger: every satisfiable formula must be satisfied by a model whose size is bounded by some function on the size of the formula. It is known that that the modal μ-calculus has the small model property; particularly, every satisfiable formula is true in a model whose size is exponential in

the length of the formula.

The first direct proof of the finite model property was by Kozen [Koz83]. The proof employs results from the theory of well-quasi ordering. Basically, the proof shows that, for any formula ϕ, any model of ϕ can be turned into a finite model. The idea is to define a well-quasi ordering which compares any two states based on the subformulae of ϕ true at those states and the least approximants of the least-fixpoint formulae which make those formulae true. A finite model for ϕ can then be obtained by taking the quotient of the starting model by such a well-quasi ordering. Hence, in a way, this approach refines the filtration method in modal logic (but which fails for the modal μ-calculus). Unfortunately, this simple method does not give a bound on the size of the finite model obtained.

The best known proof of the small model property is a consequence of the automata-theoretic method for satisfiability checking. Particularly, it follows from the *Regularity Theorem* ([Rab72], [Tho90]) which implies that if the automaton constructed from the given formula accepts some tree, it must accept a regular tree unwound from a graph whose size is exponential in the size of the formula. This approach is very indirect. To understand why the logic has the small model property, we have to go through the proofs of the related theorems in automata theory. For this reason, we have been trying to find a more direct proof of the small model property. In particular, we have been studying operations on models which can be applied to convert a model of the formula into a small model. Surprisingly, apart from the operations studied in modal logics (which are insufficient for our purpose), model-manipulation techniques for the modal μ-calculus are not common in literature.

1.4 Contributions

The contributing results in this thesis can be summarised as follows:

(1) The tableau system TS in Chapter 4 is, as far as we know, the first tableau system for satisfiability where every tableau is *finite*. The soundness and completeness of the tableau system immediately implies the decidability of the satisfiability problem and the small model property for the modal μ-calculus. The novel idea of this tableau system is the use of *names* to keep track of the unfoldings of μ-variables and the notion of *name signatures*, which are used to guide the construction of a successful tableau for a satisfiable formula in the completeness proof.

(2) The tableau system ACON for the *aconjunctive* fragment and the axiomatic completeness proof in Section 4.2. The first tableau system for this fragment of the logic was given in [Koz83]. However, we find our tableau system much cleaner and easier to understand. The tableau system TS for the full modal μ-calculus can be seen a generalisation of tableau system ACON. We are still working on extending the axiomatic completeness proof based on ACON to the full logic using tableau system TS instead.

(3) The study of model-surgery techniques in Chapter 5; in particular, the notion of *trail equivalence* on pre-models. As mentioned, the goal is to prove the small model property by applying such techniques. The research in this direction is still incomplete. So far, we have only been able to show that every linear model for a formula can be turned into a small eventually-cyclic model. This provides an alternative proof of the small model theorem for the linear-time μ-calculus. The well-known proof of this latter result uses the standard transformation of linear-time μ-calculus formulae into equivalent Büchi automata ([Var88], [AN01]), and then applies the regularity theorem for Büchi automata [Tho90].

(4) The proof of the small model property and the tableau system NUMU for Π_2^μ-formulae in Section 5.3. The Π_2^μ-fragment of the modal μ-calculus has a unique property which makes it easy to prove the small model property. In a sense, the result for this fragment can be seen as a generalisation of the techniques used in proving the small model property for the temporal logics LTL, CTL, and PDL ([Eme90], [Sti92]).

1.5 Outline

The rest of the thesis is organised as follows. In Chapter 2, we define the syntax and the semantics of the modal μ-calculus and define related terminology. Chapter 3 describes the notions of *pre-models*, *trails*, and *signatures*, and proves the *Fundamental Semantic Theorem of the Modal μ-Calculus*. In the last section of Chapter 3, we describe a general overview of tableau systems and give a simple tableau system for the modal μ-calculus based on [NW97]. In Chapter 4, we present our tableau systems TS and ACON and prove their soundness and completeness. The axiomatic completeness of the aconjunctive fragment is also given in this chapter. In Chapter 5, we describe our study of model-surgical techniques and illustrate some of their applications. The second half of Chapter 5 deals with the small model property of Π_2^μ-formulae and gives a sound and complete tableau system for the fragment. In Chapter 6, we conclude and describe future directions of our research.

Chapter 2

Modal μ-Calculus

2.1 Syntax

The *modal μ-calculus* [Koz83] can be seen as an extension of a modal logic of actions with a family of operators μX and νX, called the *least fixpoint operators* and the *greatest fixpoint operators*, respectively. *Formulae* in the modal μ-calculus are built up from the logical symbols (Boolean connectives, modal operators, and fixpoint operators) and the non-logical symbols in the following sets:

- Prop: the (countably infinite) set of *proposition letters*, ranged over by $P, Q, ...$
- Var: the (countably infinite) set of *variables*, ranged over by $Z, Y, X, ...$
- Act: the (countably infinite) set of *actions*, ranged over by $a, b, ...$

The subscript and/or superscript versions of the above symbols (for example P_i, a^j, or Z_i^j) are also used.

The language of the modal μ-calculus, denoted by $\mathcal{L}_\mu$, consists of the formulae generated from the following grammar:[1]

$$
\begin{array}{rll}
\phi ::= & P \mid X & \\
& \mid \neg\phi & \\
& \mid \phi \vee \phi & \textit{“disjunctive formulae”} \\
& \mid \phi \wedge \phi & \textit{“conjunctive formulae”} \\
& \mid \langle a \rangle \phi \mid [a]\phi & \textit{“modal formulae”} \\
& \mid \mu X.\phi & \textit{“μ-formulae”} \\
& \mid \nu X.\phi & \textit{“ν-formulae”.}
\end{array}
$$

where, in the last two cases, there is a restriction that each free occurrence of X in ϕ (i.e. an occurrence of X not within the scope of a fixpoint operator μX or νX) lies

[1] It suffices to define the language with either $\vee$ or $\wedge$, either $\langle\cdot\rangle$ or $[\cdot]$, and either μ or ν as primitives, and define the dual operators as abbreviations (e.g. $\nu X.\phi(X)$ abbreviates $\neg\mu X.\neg\phi(\neg X)$). However, on many occasions it is more suitable to treat both types of fixpoint operators as primitives (for example when defining the subformulae or the length of a formula). So we decide to make the dual operators primitives in the language. Another commonly used definition is to include only positive formulae in the language.

within an even number of negations in ϕ. As will be clear from the semantics, this syntactic restriction guarantees that the function defined by ϕ is monotone in X, and hence the least and greatest fixpoints of such function (with respect to X) always exist.

We typically use the Greek letters ϕ, ψ, γ (and their scripted versions) to range over formulae. Following [Koz83], the symbol σ is used to stand for either μ or ν. Modal formulae $\langle\cdot\rangle\phi$ and $[\cdot]\phi$ are also referred to as $\langle\cdot\rangle$-*formulae* and $[\cdot]$-*formulae*, respectively. A *literal* is either a proposition letter or its negation.

Other common operators are defined as usual: $\phi \to \psi = \neg\phi \vee \psi$, $\phi \leftrightarrow \psi = (\phi \to \psi) \wedge (\psi \to \phi)$, $\bot = P \wedge \neg P$, and $\top = P \vee \neg P$, for some proposition letter P.

To minimise the use of parentheses, we assume the following precedence of operators, from highest to lowest: $\neg, \langle a\rangle, [a], \wedge, \vee, \sigma X, \to, \leftrightarrow$. For example, we write $\mu Z.P \vee \nu Y.[a]Y \wedge [a]Z$ for $\mu Z.(P \vee \nu Y.(([a]Y) \wedge ([a]Z)))$.

Positive formulae. For convenience, in most of the thesis, we restrict ourselves to the formulae in which the negation symbol only appears next to proposition letters, called *positive formulae*. It is shown in the next section that every closed formula is semantically equivalent to a positive formula.

For the rest of this section, we describe some syntactic terminology on the formulae of the modal μ-calculus. The first one is the notion of *subformulae*. We define the set of *subformulae* of a formula ϕ, denoted by $\mathrm{Sub}(\phi)$, inductively as follows:

$$\begin{aligned}
\mathrm{Sub}(P) &= \{P\},\\
\mathrm{Sub}(X) &= \{X\},\\
\mathrm{Sub}(\neg\phi) &= \mathrm{Sub}(\phi) \cup \{\neg\phi\},\\
\mathrm{Sub}(\phi_1 \vee \phi_2) &= \mathrm{Sub}(\phi_1) \cup \mathrm{Sub}(\phi_2) \cup \{\phi_1 \vee \phi_2\},\\
\mathrm{Sub}(\phi_1 \wedge \phi_2) &= \mathrm{Sub}(\phi_1) \cup \mathrm{Sub}(\phi_2) \cup \{\phi_1 \wedge \phi_2\},\\
\mathrm{Sub}(\langle a\rangle\phi) &= \mathrm{Sub}(\phi) \cup \{\langle a\rangle\phi\},\\
\mathrm{Sub}([a]\phi) &= \mathrm{Sub}(\phi) \cup \{[a]\phi\},\\
\mathrm{Sub}(\mu X.\phi) &= \mathrm{Sub}(\phi) \cup \{X, \mu X.\phi\},\\
\mathrm{Sub}(\nu X.\phi) &= \mathrm{Sub}(\phi) \cup \{X, \nu X.\phi\}.
\end{aligned}$$

Note that we include the variable X in $\mathrm{Sub}(\sigma X.\phi)$ even though X may not occur in $\sigma X.\phi$. This is mainly for technical convenience. Another commonly-used notion of subformulae is the *(Fischer-Ladner) closure* in [Koz83],[SE89]. The key difference is that instead of considering $\phi(X)$ as a subformula of $\sigma X.\phi(X)$, its *unfolding* $\phi(\sigma X.\phi(X))$ is used instead.

The *length of* ϕ, denoted by $|\phi|$, is given as follows:

$$\begin{aligned}
|P| &= 1,\\
|X| &= 1,\\
|\neg\phi| &= |\phi| + 1,\\
|\phi_1 \vee \phi_2| &= |\phi_1| + |\phi_2| + 1,\\
|\phi_1 \wedge \phi_2| &= |\phi_1| + |\phi_2| + 1,\\
|\langle a\rangle\phi| &= |\phi| + 1,\\
|[a]\phi| &= |\phi| + 1,\\
|\mu X.\phi| &= |\phi| + 2,\\
|\nu X.\phi| &= |\phi| + 2.
\end{aligned}$$

The reason we add 2 to $|\phi|$ in the last two cases is to ensure that the following nice relationship still holds with the above definition of subformulae.

Fact 2.1. *For any formula* ϕ, $|\mathrm{Sub}(\phi)| \leq |\phi|$.

The notions of free and bound occurrences of variables are as usual: an occurrence of a variable X in ϕ is said to be *bound* iff it lies within the scope of some occurrence of a fixpoint operator σX in ϕ; the occurrence is said to be *free* otherwise. Formulae without free occurrences of variables are called *closed formulae*.

Definition 2.2. An occurrence of variable X is said to be *positive* (*negative*) iff it lies within the scope of an *even* (resp., *odd*) number of negations. A formula ϕ is said to be *positive* (*negative*) in variable X iff every free occurrence of X in ϕ is positive (resp., negative).

Note that, when we say a *variable in formula* ϕ, we always mean a variable X where either X or an operator σX has an occurrence in ϕ.

Substitution. For any formulae ϕ, ψ and any variable X, we use the term $\phi\{\psi/X\}$ to denote the formula resulted from replacing each *free occurrence* of X in ϕ by ψ, *provided that each free occurrence of any variable in* ψ *does not become bound in the process* (we say that the substitution $\{\psi/X\}$ is *safe* for ϕ if this latter condition holds). If a formula is first written as $\phi(X)$, it is to be understood that the subsequent writing of $\phi(\psi)$ denotes the term $\phi\{\psi/X\}$. Note that the term $\phi(X)$ does *not* suggest that X must occur free in ϕ nor that X is the only free variable in ϕ.

Definition 2.3 (Positive Normal Form [Koz83])**.** A formula ϕ is said to be in *positive normal form* (or *p.n.f.*) iff ϕ is a *positive* formula such that, for each variable X, there is at most one occurrence of a fixpoint operator for X and, if X occurs free in ϕ, no fixpoint operator for X occurs in ϕ.

It is obvious that the class of formulae in positive normal form is closed under subformulae. It is well known that every closed formula is semantically equivalent to one in positive normal form [Koz83].

Suppose ϕ is a formula in positive normal form. A *free variable* in ϕ is a variable which has a free occurrence in ϕ. A *bound variable* in ϕ is a variable X where σX has an occurrence in ϕ (thus a bound variable does *not* necessarily occur in ϕ). Since ϕ is in positive normal form, each variable in ϕ is either a free variable or a bound variable (but not both). For each bound variable X in ϕ, the unique subformula of ϕ of the form $\sigma X.\psi$ is said to be *identified by* X. A bound variable X is said to be of μ-type (of ν-type) or called a μ-variable (ν-variable) iff the subformula identified by X is a μ-formula (respectively ν-formula).

Ordering of variables. The bound variables in formula ϕ can be partially ordered based on the nesting of their identified fixpoint formulae. Precisely, for any bound variables X, Y in ϕ, X is said to be *higher* than Y, written $X \preccurlyeq Y$, iff the fixpoint formula identified by Y is a proper subformula of the one identified by X. We usually use the term *outermost variable* for the highest variable.

Observe that, for each subformula ψ of ϕ, the variables which occur free in ψ are *linearly ordered* under $\preccurlyeq$. Similarly, for each variable X, the set of variables higher than X is linearly ordered.

For example, in the formula

$$\nu X.(\nu Y.P \wedge \mu Z.[a]Y \vee [a]Z) \wedge \langle a\rangle \mu Z'.X \vee [a]Z',$$

X is the outermost variable in ϕ, and $X \prec Y \prec Z$ and $X \prec Z'$.

Definition 2.4 (Active Variables)**.** Suppose ϕ is in positive normal form. For any subformula ψ of ϕ, a variable X is said to be *active* in ψ iff there is a sequence $\sigma_1 X_1.\psi_1, ..., \sigma_n X_n.\psi_n$ $(n \geq 1)$ of subformulae of ϕ such that

- $X_1 = X$,
- each X_i $(i < n)$ has a free occurrence in $\sigma X_{i+1}.\psi_{i+1}$, and
- X_n has a free occurrence in ψ.

Note that the above conditions imply that $X_1 \prec ... \prec X_n$.

This notion of active variables were first defined in [Koz83]. The idea behind this notion is that, when the free occurrences of a variable in a subformula ψ of ϕ are replaced by the body of its identified fixpoint formula, a new free occurrence of variables can appear. X is considered *active* in ϕ if we can repeat this process until a formula where X occurs free is obtained. For example, consider the formula

$$\mu X.P \vee \nu Y.\langle a\rangle X \wedge \mu Z.Y \vee [a]Z.$$

X does *not* occur free in $\mu Z.Y \vee [a]Z$ but is active in it, because it occurs free in $\mu Z.(\langle a\rangle X \wedge \mu Z.Y \vee [a]Z) \vee [a]Z$.

Below are some properties of active variables.

Lemma 2.5. *Suppose variables X and Y identify $\sigma_X X.\psi_X$ and $\sigma_Y Y.\psi_Y$, respectively.*

(a) If X is active in ψ and Y is active in ψ_X then Y is active in ψ.

(b) If X and Y are both active in ψ then either X is active in ψ_Y (hence $X \preccurlyeq Y$) or Y is active in ψ_X (hence $Y \preccurlyeq X$).

Proof. This is straightforward from the definition. □

It follows that the set of active variables in a formula is linearly ordered by $\preccurlyeq$ and thus each non-empty subset of it has the least element, i.e. the outermost variable in the set.

Lemma 2.6. *For any subformula ψ of ϕ, the set of variables active in ψ is linearly ordered under $\preccurlyeq$.*

Proof. By Lemma 2.5 (b). □

2.2 Semantics

Models. A *model* in the modal μ-calculus is a pair $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}}\rangle$, where

- $\mathcal{S} = \langle M, \{R_a\}_{a\in \mathrm{Act}}\rangle$ is a *non-empty* transition system over Act,
- $\mathcal{V}_{\mathrm{Prop}} : \mathrm{Prop} \to \wp(M)$, called a *propositional valuation over* $\mathcal{S}$, is a function assigning a set of states in M to each proposition letter.

A model is sometimes written as a triple $\langle M, \{R_a\}_{a\in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}}\rangle$.

A *valuation* $\mathcal{V}$ over a model $\mathcal{M}$ is a function assigning a set of states of $\mathcal{M}$ to each variable. Formulae in the modal μ-calculus are interpreted over a model and a valuation. Given a formula ϕ, a model $\mathcal{M} = \langle M, \{R_a\}_{a\in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}}\rangle$, and a valuation $\mathcal{V}$, we denote the *set of states at which ϕ is true* by $\|\phi\|^{\mathcal{M}}_{\mathcal{V}}$. This can be defined inductively on ϕ:

$$
\begin{aligned}
\|P\|^{\mathcal{M}}_{\mathcal{V}} &= \mathcal{V}_{\mathrm{Prop}}(P),\\
\|X\|^{\mathcal{M}}_{\mathcal{V}} &= \mathcal{V}(X),\\
\|\neg\phi\|^{\mathcal{M}}_{\mathcal{V}} &= M - \|\phi\|^{\mathcal{M}}_{\mathcal{V}},\\
|\phi_1 \vee \phi_2\|^{\mathcal{M}}_{\mathcal{V}} &= \|\phi_1\|^{\mathcal{M}}_{\mathcal{V}} \cup \|\phi_2\|^{\mathcal{M}}_{\mathcal{V}},\\
\|\phi_1 \wedge \phi_2\|^{\mathcal{M}}_{\mathcal{V}} &= \|\phi_1\|^{\mathcal{M}}_{\mathcal{V}} \cap \|\phi_2\|^{\mathcal{M}}_{\mathcal{V}},\\
\|\langle a\rangle\phi\|^{\mathcal{M}}_{\mathcal{V}} &= \{s \in M \mid \exists t.sR_at, t \in \|\phi\|^{\mathcal{M}}_{\mathcal{V}}\},\\
\|[a]\phi\|^{\mathcal{M}}_{\mathcal{V}} &= \{s \in M \mid \forall t.sR_at \to t \in \|\phi\|^{\mathcal{M}}_{\mathcal{V}}\},\\
\|\mu X.\phi\|^{\mathcal{M}}_{\mathcal{V}} &= \bigcap\{S \subseteq M \mid \|\phi\|^{\mathcal{M}}_{\mathcal{V}[X:=S]} \subseteq S\},\\
\|\nu X.\phi\|^{\mathcal{M}}_{\mathcal{V}} &= \bigcup\{S \subseteq M \mid S \subseteq \|\phi\|^{\mathcal{M}}_{\mathcal{V}[X:=S]}\}.
\end{aligned}
$$

where $\mathcal{V}[X := S]$ is the valuation in which $\mathcal{V}[X := S](X) = S$ and $\mathcal{V}[X := S](Y) = \mathcal{V}(Y)$ for each variable Y other than X. Superscript $\mathcal{M}$ and subscript $\mathcal{V}$ are omitted whenever possible.

A formula ϕ is said to be *satisfied by model $\mathcal{M}$ and valuation $\mathcal{V}$ at state s*, written $\mathcal{M}, s \models_{\mathcal{V}} \phi$, iff $s \in \|\phi\|^{\mathcal{M}}_{\mathcal{V}}$. Similarly, a set Γ of formulae is satisfied by $\mathcal{M}$ and $\mathcal{V}$ at s, written $\mathcal{M}, s \models_{\mathcal{V}} \Gamma$, iff $\mathcal{M}, s \models_{\mathcal{V}} \phi$ for each $\phi \in \Gamma$. If we only consider a closed formula ϕ or a set of closed formulae Γ, we omit the subscript $\mathcal{V}$.

A formula or a set of formulae is said to be *satisfiable* iff there is a model, a valuation, and a state satisfying it. A formula ϕ is said to be *valid*, written $\models \phi$, iff ϕ is true at every state in every model under any valuation. Two formulae ϕ, ψ are said to be *semantically equivalent* iff $\models \phi \leftrightarrow \psi$.

From the definition, $\|\mu X.\psi\|_{\mathcal{V}}$ is defined as the g.l.b. of all the pre-fixpoints of the function $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$. We can show that $\|\mu X.\psi\|_{\mathcal{V}}$ is indeed the least fixpoint of this latter function. First we must show that the function is monotone. This follows from the following proposition (and the assumption that ψ must be positive in X).

Proposition 2.7 (Monotonicity)**.** *For any formula ϕ and variable X,*

(a) if ϕ is positive *in X then $\|\phi\|_{\mathcal{V}[X:=S]} \subseteq \|\phi\|_{\mathcal{V}[X:=S']}$ for any sets $S \subseteq S'$;*
(b) if ϕ is negative *in X then $\|\phi\|_{\mathcal{V}[X:=S']} \subseteq \|\phi\|_{\mathcal{V}[X:=S]}$ for any sets $S \subseteq S'$.*

Proof. We use induction on ϕ to show that (a) and (b) hold (for any variable X). The cases where ϕ is a proposition letter or a variable are obvious. Suppose $\phi = \neg\psi$. If ϕ is positive in X then ψ must be negative in X. By induction, for any sets $S \subseteq S'$, $\|\psi\|_{\mathcal{V}[X:=S']} \subseteq \|\psi\|_{\mathcal{V}[X:=S]}$, which implies that $\|\neg\psi\|_{\mathcal{V}[X:=S]} \subseteq \|\neg\psi\|_{\mathcal{V}[X:=S']}$. The case where ϕ is negative in X can be shown similarly. Other non-fixpoint cases are straightforward.

Suppose $\phi = \sigma Y.\psi$. The statements obviously hold if $X = Y$, so we assume otherwise. Let $\|\psi\|(S,T)$ denote $\|\psi\|_{\mathcal{V}[X:=S][Y:=T]}$. Suppose ϕ is positive in X; thus so is ψ. By induction, $\|\psi\|(S,T) \subseteq \|\psi\|(S',T)$ for any sets S, S', T where $S \subseteq S'$. Thus, for $\sigma = \mu$,

$$\begin{aligned} \|\mu Y.\psi\|_{\mathcal{V}[X:=S]} &= \bigcap\{T \subseteq M \mid \|\psi\|(S,T) \subseteq T\} \\ &\subseteq \bigcap\{T \subseteq M \mid \|\psi\|(S',T) \subseteq T\} \\ &= \|\mu Y.\psi\|_{\mathcal{V}[X:=S']}, \end{aligned}$$

for any sets $S \subseteq S'$. Similarly, for $\sigma = \nu$,

$$\begin{aligned} \|\nu Y.\psi\|_{\mathcal{V}[X:=S]} &= \bigcup\{T \subseteq M \mid T \subseteq \|\psi\|(S,T)\} \\ &\subseteq \bigcup\{T \subseteq M \mid T \subseteq \|\psi\|(S',T)\} \\ &= \|\nu Y.\psi\|_{\mathcal{V}[X:=S']}, \end{aligned}$$

for any sets $S \subseteq S'$. The case where ϕ is negative in X can be shown similarly. □

By Knaster-Tarski Theorem (Theorem 0.1), it is then equivalent to define $\|\mu X.\psi\|_{\mathcal{V}}$ as the least fixpoint of $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$. Further, Theorem 0.2 tells us that the semantics can be given by an iterative process. To state this precisely, auxiliary formulae $\mu^{\alpha} X.\psi$ (where α ranges over ordinals) called the *approximants of* $\mu X.\psi$ are introduced. The semantics of these formulae can be given as follows:

$$\begin{aligned} \|\mu^0 X.\psi\|_{\mathcal{V}} &= \emptyset, \\ \|\mu^{\alpha+1} X.\psi\|_{\mathcal{V}} &= \|\psi\|_{\mathcal{V}[X:=\|\mu^{\alpha} X.\psi\|_{\mathcal{V}}]}, \\ \|\mu^{\lambda} X.\psi\|_{\mathcal{V}} &= \bigcup_{\alpha<\lambda} \|\mu^{\alpha} X.\psi\|_{\mathcal{V}}, \end{aligned}$$

where α denotes an ordinal and λ a limit ordinal. We can thus summarise the semantics of the least fixpoint formulae as follows:

Proposition 2.8. *Let f be $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$.*

$$\|\mu X.\psi\|_{\mathcal{V}} = \bigcap\{S \subseteq M \mid f(S) \subseteq S\} = \mu f = \textstyle\bigcup_{\alpha \in \mathbb{O}} \|\mu^{\alpha} X.\psi\|_{\mathcal{V}}.$$

Proof. The first equation is the semantics of $\mu X.\psi$. Since f is a monotone function (by Proposition 2.7), we obtain the second equation using Theorem 0.1. The last equation follows from Theorem 0.2. □

The operators νX can be treated dually. $\|\nu X.\psi\|_{\mathcal{V}}$ is defined as the l.u.b. of all the post-fixpoints of the function $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$, which by Knaster-Tarski Theorem and the monotonicity of the function, is equal to its greatest fixpoint. The approximants for $\nu X.\psi$, written $\nu^{\alpha} X.\psi$ (where α ranges over ordinals), can be given similarly:

$$\begin{aligned} \|\nu^0 X.\psi\|_{\mathcal{V}} &= M, \\ \|\nu^{\alpha+1} X.\psi\|_{\mathcal{V}} &= \|\psi\|_{\mathcal{V}[X:=\|\nu^{\alpha} X.\psi\|_{\mathcal{V}}]}, \\ \|\nu^{\lambda} X.\psi\|_{\mathcal{V}} &= \bigcap_{\alpha<\lambda} \|\nu^{\alpha} X.\psi\|_{\mathcal{V}}, \end{aligned}$$

where α denote an ordinal and λ a limit ordinal. The semantics of the greatest fixpoint formulae can be summarised as follows:

Proposition 2.9. *Let f be $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$.*

$$\|\nu X.\psi\|_{\mathcal{V}} = \bigcup\{S \subseteq M \mid S \subseteq f(S)\} = \nu f = \textstyle\bigcap_{\alpha \in \mathbb{O}} \|\nu^{\alpha} X.\psi\|_{\mathcal{V}}.$$

Proof. The proof is similar to the least fixpoint case. □

One way to evaluate a fixpoint formula $\sigma X.\psi$ in a model is by successively computing the approximations $\|\sigma^0 X.\psi\|, \|\sigma^1 X.\psi\|, ...$ On a finite model of size n, this sequence will eventually converge to $\|\sigma X.\psi\|$ at $\|\sigma^i X.\psi\|$ for some $i \leq n$. For infinite models, the sequence may not converge at any finite approximation. Moreover, since the function

$\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$ may not be continuous, we may need to go beyond $\|\sigma^\omega X.\psi\|$ even on a countable model. Here are some examples.

Example 2.10. Let $\mathcal{M}$ be the model depicted in Figure 2.1.

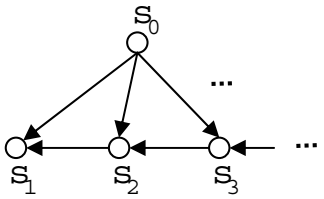


Figure 2.1: Model $\mathcal{M}$ in Example 2.10.

The set $\|\mu X.[a]X\|$ is equal to $\|\mu^{\omega+1}X.[a]X\|$ as shown below.

$$\begin{aligned}
\|\mu^0 X.[a]X\| &= \emptyset, \\
\|\mu^i X.[a]X\| &= \{s_1, ..., s_i\}, 1 \leq i < \omega, \\
\|\mu^\omega X.[a]X\| &= \{s_1, s_2, ...\}, \\
\|\mu^{\omega+1} X.[a]X\| &= \|\mu^{\omega+2} X.[a]X\| = \{s_0, s_1, s_2, ...\}.
\end{aligned}$$

Example 2.11. Let $\mathcal{M}$ be the model depicted in Figure 2.2.

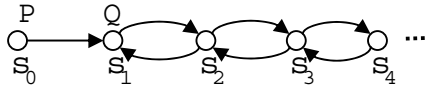


Figure 2.2: Model $\mathcal{M}$ in Example 2.11. P, Q are true only at indicated states.

Consider the formula

$$\mu X.\phi \quad = \quad \mu X.Q \vee (\neg P \wedge \langle a \rangle X) \vee \langle a \rangle \nu Y.X \wedge [a]Y.$$

$\|\mu X.\phi\|$ can be computed as follows.

$$\begin{aligned}
\|\mu^0 X.\phi\| &= \emptyset, \\
\|\mu^i X.\phi\| &= \{s_1, ..., s_i\}, 1 \leq i < \omega, \\
\|\mu^\omega X.\phi\| &= \{s_1, s_2, ...\}, \\
\|\mu^{\omega+1} X.\phi\| &= \|\mu^{\omega+2} X.\phi\| = \{s_0, s_1, s_2, ...\}.
\end{aligned}$$

The function $\lambda S.\|\nu Y.X \wedge [a]Y\|_{\mathcal{V}[X:=S]}$ is *not* continuous, as seen below.

$$\begin{aligned}\|\nu Y.X \wedge [a]Y\|_{\mathcal{V}[X:=\|\mu^i X.\phi\|]} &= \emptyset, 0 \leq i < \omega,\\ \|\nu Y.X \wedge [a]Y\|_{\mathcal{V}[X:=\|\mu^\omega X.\phi\|]} &= \{s_1, s_2, ...\},\\ \|\nu Y.X \wedge [a]Y\|_{\mathcal{V}[X:=\|\mu^{\omega+1} X.\phi\|]} &= \{s_0, s_1, s_2, ...\}.\end{aligned}$$

Besides its practical purpose, the iterative semantics of fixpoint formulae is useful for proving properties of the logic. Particularly, it allows us to use an inductive argument on approximants. The following proposition is a simple consequence of the above discussion. Roughly, it states that if a least fixpoint formula is true at some state, then it has a *least* approximant true at that state (and dually for a greatest fixpoint formula). The least approximants are a basis for the notion of *signatures* [SE89], which is of great importance in the modal μ-calculus. We explain signatures in the next chapter.

Proposition 2.12.

(a) $\mathcal{M}, s \models_{\mathcal{V}} \mu X.\psi$ *iff there exists an ordinal* β *such that* $\mathcal{M}, s \models_{\mathcal{V}} \mu^\beta X.\psi$ *and, for all ordinals* $\alpha < \beta$, $\mathcal{M}, s \not\models_{\mathcal{V}} \mu^\alpha X.\psi$.

(b) $\mathcal{M}, s \not\models_{\mathcal{V}} \nu X.\psi$ *iff there exists an ordinal* β *such that* $\mathcal{M}, s \not\models_{\mathcal{V}} \nu^\beta X.\psi$ *and, for all ordinals* $\alpha < \beta$, $\mathcal{M}, s \models_{\mathcal{V}} \nu^\alpha X.\psi$.

Proof. This is a simple consequence of Proposition 2.8 and 2.9. □

We now turn to study some useful semantic properties.

Proposition 2.13. *Suppose* $\phi(X), \psi$ *are any formulae and* $\phi(X)$ *is positive in* X.

(a) $\models \langle a\rangle\psi \leftrightarrow \neg[a]\neg\psi$.

(b) $\models \nu X.\phi(X) \leftrightarrow \neg\mu X.\neg\phi(\neg X)$.

(c) $\models \sigma X.\phi(X) \leftrightarrow \sigma Y.\phi(Y)$, *provided that* Y *does* not *occur free in* $\phi(X)$.

(d) $\models \phi(\sigma X.\phi(X)) \leftrightarrow \sigma X.\phi(X)$.

(e) $\models \phi(\psi) \rightarrow \psi$ *implies* $\models \mu X.\phi(X) \rightarrow \psi$.

(f) $\models \psi \rightarrow \phi(\psi)$ *implies* $\models \psi \rightarrow \nu X.\phi(X)$.

(g) $\models \psi(X) \rightarrow \phi(X)$ *implies* $\models \sigma X.\psi(X) \rightarrow \sigma X.\phi(X)$, *provided that* $\psi(X), \phi(X)$ *are positive in* X.

Proof. This is straightforward to check from the semantics. □

It is obvious from the semantics that a formula can be turned into the positive normal form by applying operator dualities and renaming bound variables. Note that since negated variables (e.g. $\neg X$) are not included in our definition of the positive normal form, this only applies to closed formulae.

Proposition 2.14 ([Koz83])**.** *Every closed formula* ϕ *is semantically equivalent to a formula in positive normal form of length linear in* $|\phi|$.

Proof. A closed formula can be converted into an equivalent positive formula by subsequently applying the following rules:

$$\begin{aligned}
\neg\neg\phi &\Rightarrow \phi \\
\neg(\phi \vee \psi) &\Rightarrow \neg\phi \wedge \neg\psi \\
\neg(\phi \wedge \psi) &\Rightarrow \neg\phi \vee \neg\psi \\
\neg(\langle a\rangle\phi) &\Rightarrow [a]\neg\phi \\
\neg([a]\phi) &\Rightarrow \langle a\rangle\neg\phi \\
\neg(\mu X.\phi(X)) &\Rightarrow \nu X.\neg\phi(\neg X) \\
\neg(\nu X.\phi(X)) &\Rightarrow \mu X.\neg\phi(\neg X)
\end{aligned}$$

By Proposition 2.13 (and simple propositional equivalences), the formula resulted from applying any of these rules is semantically equivalent to the original formula. A positive formula can be turned into positive normal form by renaming bound variables. □

Definition 2.15 (Guarded Formulae [Wal93])**.** An occurrence of a variable X is said to be *guarded* in a formula ψ iff the occurrence lies within a modal subformula of ψ. Formula ϕ is said to be *guarded* iff, for each fixpoint subformula $\sigma X.\psi$ of ϕ, every free occurrence of X in ψ is guarded in ψ. We use the term *unguarded* for 'not guarded'.

For example, consider the formula $\phi = \langle a\rangle X \vee \nu Y.X \wedge [a]Y$. The first occurrence of X is guarded in ϕ, whereas the second one is not. Hence, $\mu X.\phi$ is not a guarded formula. The formula $\nu Y.X \wedge [a]Y$ is guarded.

Every formula can be translated into a semantically equivalent guarded formula. This has been shown in [Wal93], [KVW00], and [Mat02]. The proofs in these papers employ the following semantic properties.

Lemma 2.16. *For any formulae $\phi(X), \psi(X)$ positive in X,*

(a) $\models \mu X.(X \vee \psi(X)) \wedge \phi(X) \leftrightarrow \mu X.\psi(X) \wedge \phi(X)$.
(b) $\models \nu X.(X \wedge \psi(X)) \vee \phi(X) \leftrightarrow \nu X.\psi(X) \vee \phi(X)$.

Proof. (a) Suppose $\alpha(X) = (X \vee \psi(X)) \wedge \phi(X)$ and $\beta(X) = \psi(X) \wedge \phi(X)$.

$\models \alpha(X) \rightarrow X \vee \beta(X)$.
$\models \alpha(\mu X.\beta(X)) \rightarrow \mu X.\beta(X) \vee \beta(\mu X.\beta(X))$.
$\models \alpha(\mu X.\beta(X)) \rightarrow \mu X.\beta(X)$ (by Proposition 2.13(d)).
$\models \mu X.\alpha(X) \rightarrow \mu X.\beta(X)$ (by Proposition 2.13(e)).

The other direction follows from $\models \beta(X) \rightarrow \alpha(X)$ and Proposition 2.13(g).

(b) is dual. □

Lemma 2.17. *Suppose $\phi(X, Y)$ is a formula which is positive in X and Y, and where each free occurrence of X is* unguarded *and does* not *lie within the scope of any fixpoint*

operator. Then

$$\models \mu Y.\phi(Y,Y) \leftrightarrow \mu Y.\phi(\bot,Y)$$
$$\models \nu Y.\phi(Y,Y) \leftrightarrow \nu Y.\phi(\top,Y)$$

Proof. By writing $\phi(X,Y)$ in CNF, it can be shown that

$$\models \phi(X,Y) \leftrightarrow (X \vee \alpha(Y)) \wedge \beta(Y),$$

for some formulae $\alpha(Y), \beta(Y)$ *not* containing occurrences of X. This implies that $\models \phi(\bot,Y) \leftrightarrow \alpha(Y) \wedge \beta(Y)$. Hence, by Proposition 2.13(g), $\models \mu Y.\phi(\bot,Y) \leftrightarrow \mu Y.\alpha(Y) \wedge \beta(Y)$. From Lemma 2.16(a), $\mu Y.\phi(Y,Y) \leftrightarrow \mu Y.\alpha(Y) \wedge \beta(Y)$. Thus, $\models \mu Y.\phi(Y,Y) \leftrightarrow \mu Y.\phi(\bot,Y)$ as required. The ν-case is similar. □

In the previous lemma, the condition that each free occurrence of X is *not* in the scope of a fixpoint operator is necessary. For example, the formula $\mu X.P \vee \nu Y.X \vee \langle a \rangle Y$ is clearly not equivalent to $\mu X.P \vee \nu Y.\langle a \rangle Y$.

Proposition 2.18 ([Wal93], [KVW00], [Mat02])**.** *Every formula is semantically equivalent to a guarded one.*

Proof. We shall use the transformation of a formula into a guarded equivalent described in [KVW00]. Suppose ϕ is a formula, which is assumed w.l.o.g. to be well-named. First, for any formula ψ and variable X, we define $f(\psi,\mu,X)$ (resp., $f(\psi,\nu,X)$) to be the formula obtained from ψ by replacing each occurrence of X that is not within the scope of any modal or fixpoint operator by $\bot$ (resp., $\top$). Applying the previous lemma, we may deduce that, for any formula $\sigma X.\psi$,

$$\models \sigma X.\psi \leftrightarrow \sigma X.f(\psi,\sigma,X).$$

For brevity, let Unfold$(\sigma X.\psi)$ denote $\psi\{\sigma X.\psi/X\}$.

To transform ϕ into a guarded formula, we proceed by subsequently replacing each fixpoint formula $\sigma X.\psi$ in ϕ by either $\sigma X.f(\psi,\sigma,X)$ or Unfold$(\sigma X.f(\psi,\sigma,X))$, starting from an *innermost* variable X. Precisely, suppose $X_1,...,X_m$ are the variables in ϕ such that X_i *lower* than X_j implies $i < j$. Define the formulae $\phi_0,...,\phi_m$ as follows:

- $\phi_0 = \phi$.
- For each i, $0 < i \leq m$, suppose X_i identifies the formula $\sigma_i X_i.\psi_i$ in ϕ_{i-1}.
  - **(a)** If X_i is an outermost variable in ϕ_{i-1}, then ϕ_i is ϕ_{i-1} with $\sigma_i X_i.\psi_i$ replaced by $\sigma_i X_i.f(\psi_i,\sigma_i,X_i)$;
  - **(b)** Otherwise, ϕ_i is ϕ_{i-1} with $\sigma_i X_i.\psi_i$ replaced by Unfold$(\sigma_i X_i.f(\psi_i,\sigma_i,X_i))$.

Observe that each ϕ_i may not be well-named because Unfold$(\sigma_i X_i.f(\psi_i,\sigma_i,X_i))$ may contain more than one occurrence of operator $\sigma_j X_j$ where $j \leq i$. But since ϕ is well-

named and $X_1, ..., X_m$ are ordered such that lower variables appear earlier, we can be sure that ϕ_i will contain a unique formula $\sigma_k X_k.\psi_k$ for each $k > i$.

It is not difficult to show by induction that, for each $i \leq m$,

- for any $j \leq i$, each fixpoint formula bounded by $\sigma_j X_j$ in ϕ_i is guarded,
- for any $k > i$, if $\sigma_k X_k.\psi$ is the formula identified by X_k in ϕ_i, each unguarded occurrence of X_k in ψ does not lie within the scope of an operator $\sigma_j X_j$, for each $j \leq i$,
- ϕ_i is equivalent to ϕ (because Unfold$(\sigma X.f(\psi, \sigma, X))$, $\sigma X.f(\psi, \sigma, X)$, and $\sigma X.\psi$ are all equivalent).

Thus, ϕ_m is a guarded equivalent of ϕ. □

The transformation into guarded form described in the proof may involve repeated substitutions of a fixpoint formula $\sigma X.\psi$ by Unfold$(\sigma X.f(\psi, \sigma, X))$. Observe that, although the length of $\sigma X.f(\psi, \sigma, X)$ is no greater than that of $\sigma X.\psi$, the length of its unfolding may be quadratic in the length of $\sigma X.\psi$ (e.g. consider the unfolding of the formula $\sigma X.\langle a_1\rangle X \wedge ... \wedge \langle a_n\rangle X$). Since we repeat this operation on the given formula for each variable in it, the resulting formula may become exponentially longer than ϕ (e.g. consider the transformation of the formula $\sigma_n X_n ... \sigma_1 X_1. \bigvee_{i=1}^{n}(X_i \wedge \langle a_1\rangle X_i \wedge ... \wedge \langle a_n\rangle X_i)$). But if instead of using length we use the number of formulae in the *(Fischer-Ladner) closure* of the formula to measure its size (see [Koz83] for the definition of the closures of modal μ-calculus formulae), [KVW00] shows that the size of the resulting formula is linear in the size of the original formula. The proof uses the fact that the closure of the unfolding of a fixpoint formula is included in the closure of the fixpoint formula itself.

As far as we know, the best known upper bound on the *length* of an equivalent guarded formula is given by Mateescu [Mat02]. Mateescu uses an algorithm which is a slight improvement over [KVW00]'s algorithm described above. His analysis gives an exponential upper bound (precisely, $|g(\phi)| \leq |\phi|^{O(|\phi|)}$, where $g(\phi)$ is the formula obtained from his algorithm). He also shows that if the size of a formula is measured by the number of distinct subformulae, a quadratic upper bound can be obtained (precisely, $|\text{Sub}(g(\phi))| \leq O(|\text{Sub}(\phi)|^2)$). The proof employs the fact that the number of subformulae of the unfolding of a fixpoint formula is linear in the number of subformulae of the original formula. However, if we insist on using length as the measure, it is still not known whether the exponential blow-up is avoidable.

2.3 Basic Invariance Results

It is well-known in modal logic that modal formulae are invariant under certain operations on models [BdV01]. One basic operation is taking the *disjoint union* of a family of models.

Definition 2.19 (Disjoint Union)**.** Let $\mathcal{S}^i = \langle S^i, \{R_a^i\}_{a \in \text{Act}}\rangle$, $i \in I$, be a family of

disjoint transition systems. The *disjoint union* of $\mathcal{S}^i$, $i \in I$, denoted $\biguplus_{i\in I} \mathcal{S}^i$, is $\langle \bigcup_{i\in I} S^i, \{\bigcup_{i\in I} R^i_a\}_{a\in \mathrm{Act}}\rangle$.

Let $\mathcal{M}^i = \langle \mathcal{S}^i, \mathcal{V}^i_{\mathrm{Prop}}\rangle$, $i \in I$, be a family of *disjoint* models. The *disjoint union* of $\mathcal{M}^i$, $i \in I$, denoted $\biguplus_{i\in I} \mathcal{M}^i$, is the model $\langle \biguplus_{i\in I} \mathcal{S}^i, \mathcal{V}_{\mathrm{Prop}}\rangle$ where $\mathcal{V}_{\mathrm{Prop}}(P) = \bigcup_{i\in I} \mathcal{V}^i_{\mathrm{Prop}}(P)$ for each P.

Due to the locality of the semantics of the modal operators, it is quite obvious that this operation preserves the satisfaction of modal formulae. This is also the case for modal μ-calculus formulae.

Proposition 2.20. *Let $\mathcal{M}^i$, $i \in I$, be a family of disjoint models. For each $i \in I$, closed formula ϕ, state s in $\mathcal{M}^i$, $\mathcal{M}^i, s \models \phi$ iff $\biguplus_{i\in I} \mathcal{M}^i, s \models \phi$.*

Proof. This can be shown in the same way as for modal logic [BdV01]. □

Conversely, given any model $\mathcal{M}$ and a state s, if we remove all the states *not* reachable from s, the formulae true at s in the old model is also true in the new model, and vice versa. The model obtained in this way is called a *generated submodel* of $\mathcal{M}$ [BdV01].

Definition 2.21 (Generated Submodels)**.** Let $\mathcal{S} = \langle S, \{R_a\}_{a\in \mathrm{Act}}\rangle$ be a transition system. For any state s, the *subsystem of $\mathcal{S}$ generated by s*, denoted $\mathrm{Sub}_s(\mathcal{S})$, is $\langle S', \{R'_a\}_{a\in \mathrm{Act}}\rangle$ where S' contains s and all states to which there is a path from s, and each R'_a is the restriction R_a to S'.

Let $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}}\rangle$ be a model. The *submodel of $\mathcal{M}$ generated by state s*, denoted $\mathrm{Sub}_s(\mathcal{M})$, is the model $\langle \mathrm{Sub}_s(\mathcal{S}), \mathcal{V}'_{\mathrm{Prop}}\rangle$ where $\mathcal{V}'_{\mathrm{Prop}}$ is the restriction of $\mathcal{V}_{\mathrm{Prop}}$ to the states in $\mathrm{Sub}_s(\mathcal{S})$.

Proposition 2.22. *Given any model $\mathcal{M}$ and state s, $\mathcal{M}, s' \models \phi$ iff $\mathrm{Sub}_s(\mathcal{M}), s' \models \phi$ for any state s' in $\mathrm{Sub}_s(\mathcal{M})$ and closed formula ϕ.*

Proof. This can be shown in the same way as for modal logic [BdV01]. □

The most important invariance result is that modal formulae are invariant under bisimulation. This generalises nicely to formulae in the modal μ-calculus. Let us first recap the definition of bisimulation.

Definition 2.23 (Bisimulations)**.** Suppose $\mathcal{M} = \langle M, \{R_a\}_{a\in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}}\rangle$ and $\mathcal{M}' = \langle M', \{R'_a\}_{a\in \mathrm{Act}}, \mathcal{V}'_{\mathrm{Prop}}\rangle$ are models. A *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ is a relation $B \subseteq M \times M'$ such that whenever sBs'

(1) $s \in \mathcal{V}_{\mathrm{Prop}}(P)$ iff $s' \in \mathcal{V}'_{\mathrm{Prop}}(P)$ for each proposition letter P;
(2a) if sR_at then, for some t', $s'R_at'$ and tBt';
(2b) if $s'R_at'$ then, for some t, sR_at and tBt'.

A state s in $\mathcal{M}$ is said to be *bisimilar* to state s' in $\mathcal{M}'$, written $(\mathcal{M}, s) \cong (\mathcal{M}', s')$, iff there is a bisimulation B between $\mathcal{M}$ and $\mathcal{M}'$ such that sBs'.

Proposition 2.24. *For any state s in a model $\mathcal{M}$ and state s' in model $\mathcal{M}'$, if $(\mathcal{M}, s) \cong (\mathcal{M}', s')$ then $\mathcal{M}, s \models \phi$ iff $\mathcal{M}', s' \models \phi$, for any closed formula ϕ.*

Proof. The proof is similar to that for modal logic. See [Sti00]. □

One consequence of bisimulation invariance is that every model can be unravelled into a tree model satisfying the same formulae. By a tree model, we mean a model whose underlying transition system is a tree transition system (see Definition 0.5). The unravelling can be defined formally as follows.

Definition 2.25 (Unravelling)**.** Suppose $\mathcal{M} = \langle M, \{R_a\}_{a \in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}} \rangle$ is model and s is a state. The *unravelling* of $\mathcal{M}$ at s is the tree model $\mathcal{M}' = \langle M', \{R'_a\}_{a \in \mathrm{Act}}, \mathcal{V}'_{\mathrm{Prop}} \rangle$ where

- M' contains all sequences $s_1 ... s_n$ $(n \geq 1)$ such that $s_1 = s$ and $s_i R_{a_i} s_{i+1}$ for some $a_i \in \mathrm{Act}$ and each $i < n$,
- $\pi R_a \pi'$ iff $\pi' = \pi t$ and $\mathrm{Last}(\pi) R_a t$ (where $\mathrm{Last}(\pi)$ denotes the last state in π),
- $\mathcal{V}'_{\mathrm{Prop}}(P) = \{\pi \in M' \mid \mathrm{Last}(\pi) \in \mathcal{V}_{\mathrm{Prop}}(P)\}$.

It is clear that the state s in $\mathcal{M}$ is bisimilar to the root of the unravelling of $\mathcal{M}$ at s. Hence they satisfy the same formulae.

Lemma 2.26. *Suppose $\mathcal{M}$ is a model and $\mathcal{M}'$ is the unravelling of $\mathcal{M}$ at some state s. Then $\mathcal{M}, s \models \phi$ iff $\mathcal{M}', s \models \phi$ for any closed formula ϕ.*

Proof. Define a relation $B \subseteq M \times M'$ which include all pairs $(s_n, s_1 ... s_n)$. It is clear from the definition of unravelling that B is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. The lemma follows from Proposition 2.24. □

From this lemma, we immediately obtain the tree model property.

Proposition 2.27. *Every satisfiable formula has a tree model.*

Proof. By the previous lemma, if a formula ϕ is true at a state s in a model $\mathcal{M}$, then it is true at the root of the unravelling of $\mathcal{M}$ at s. □

In fact, it can be shown that every satisfiable formula has a tree model with a bounded degree. This is explained in the next chapter.

2.4 Alternation

One fundamental question of the modal μ-calculus is whether the alternation of the μ operators and the ν operators gives the logic more expressive power. It has been shown that this is the case. The alternation of fixpoint operators is what makes the modal μ-calculus a very expressive logic. On the other hand, it is this alternation which makes the modal μ-calculus computationally harder than other simpler temporal logics.

There are several definitions of the alternation hierarchy. The simplest one is given based on a longest sequence $\mu X_1.\psi_1, \nu X_2.\psi_2, ...$ of nested fixpoint subformulae in the formula (for example, $\mu X.(\nu Y.[a]Y) \vee \langle a \rangle X$ is higher in the hierarchy than $\mu X.P \vee \langle a \rangle X$). This approach is too coarse because it does not capture the true alternation of dependent fixpoints. In the previous example, there is clearly no dependency between $\nu Y.[a]Y$ and the outer fixpoint, and hence we should not count $\mu X, \nu Y$ as a true alternation. In fact, it is no harder to model check the former formula than a formula with only one fixpoint variable. A definition which takes such dependency into account was proposed by Emerson and Lei [EL86], and was later refined by Niwiński [Niw86]. The definition below follows that of Niwiński. Note that the alternation classes are normally defined for positive formulae.

Definition 2.28 (Alternation)**.** The classes Σ_n^μ and Π_n^μ of *positive* formulae are defined inductively as follows. $\Sigma_0^\mu = \Pi_0^\mu$ contains all the formulae without fixpoint operators. For each $n \geq 0$, Σ_{n+1}^μ (Π_{n+1}^μ) is the smallest set of formulae which contains $\Sigma_n^\mu \cup \Pi_n^\mu$ and such that

(1) if ϕ_1, ϕ_2 are in Σ_{n+1}^μ (resp. Π_{n+1}^μ), then so are $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$, $\langle a \rangle \phi_1$, and $[a]\phi_1$;
(2) if ϕ is in Σ_{n+1}^μ (resp. Π_{n+1}^μ), then so is $\mu X.\phi$ (resp. $\nu X.\phi$);
(3) if $\phi(X), \psi$ are in Σ_{n+1}^μ (resp. Π_{n+1}^μ), then so is $\phi(\psi)$ provided that no free occurrence of variables in ψ is captured by a fixpoint operator in ϕ.

The alternation depth of a formula ϕ is the least n such that $\phi \in \Sigma_{n+1}^\mu \cap \Pi_{n+1}^\mu$. The formulae of alternation depth 1 are called *alternation-free* formulae.

The following is a simple observation of the formulae in each alternation class Σ_n^μ and Π_n^μ based on the alternation of active variables in subformulae.

Lemma 2.29. *For any formula ϕ in* positive normal form*, if $\phi \in \Sigma_n^\mu$ (Π_n^μ) then for any subformula γ, the variables active in γ can be ordered as follows*

$$X_1^1 \prec ... \prec X_{k_1}^1 \prec ... \prec X_1^n \prec ... \prec X_{k_n}^n,$$

where each $k_i \geq 0$ and $X_1^i, ..., X_{k_i}^i$ are of μ-type (resp. ν-type) if i is odd and of ν-type (resp. μ-type) if i is even.

Proof. The proof is a straightforward induction on n. □

We consider some examples. The 'always eventually' formula

$$\nu X.(\mu Y.P \vee \langle a \rangle Y) \wedge [a]X$$

is in both Π_2^μ and Σ_2^μ, hence is an alternation-free formula. Π_2^μ contains formulae with $\nu\mu$ alternation, such as the 'infinitely often' formula

$$\nu X.\mu Y.((P \vee \langle a \rangle Y) \wedge [a]X).$$

The classes of alternation-free formulae, Σ_2^μ, and Π_2^μ can be defined in a more direct way as follows.

Proposition 2.30. *Assuming the formulae considered are in positive normal form.*

(a) A formula is alternation-free *iff for any subformulae $\mu X.\psi_X$ and $\nu Y.\psi_Y$, X does* not *occur free in ψ_Y and Y does* not *occur free in ψ_X.*

(b) A formula is in Σ_2^μ iff for any subformulae $\mu X.\psi_X$ and $\nu Y.\psi_Y$, Y does not *occur free in ψ_X.*

(c) A formula is in Π_2^μ iff for any subformulae $\mu X.\psi_X$ and $\nu Y.\psi_Y$, X does not *occur free in ψ_Y.*

Proof. This is straightforward from the definition. □

Lenzi [Len96] and Bradfield [Bra97] independently proved that the above alternation hierarchy is strict. Bradfield later presented a simpler proof in [Bra98] which, at the same time, provided some simple examples of strict formulae. For example, it was shown in the paper that the following formula in Σ_n^μ

$$\mu X_n.\nu X_{n-1}...\mu X_1.[c]X_1 \vee \langle a_1\rangle X_1 \vee ...\langle a_n\rangle X_n$$

is *not* semantically equivalent to any formula in Π_n^μ (or any lower alternation class).

Theorem 2.31. *For each n, there is a formula ϕ of alternation depth n which is not equivalent to any formula of alternation depth $m < n$.*

Proof. This is proved in [Len96], [Bra97], and [Bra98]. □

2.5 Axiomatisation

When the modal μ-calculus was first introduced by Kozen [Koz83], a simple axiomatisation was proposed. It was proved sound and complete for a fragment of the logic called *aconjunctive formulae*. Despite its simple form, the question whether it is complete for the full logic remained open for many years before it was affirmed by Waluekiewicz ([Wal95],[Wal00]). Before the completeness proof was found, a number of alternative deductive systems were proposed, such as in [Koz86] or [Wal93]. But arguably none of those systems are as simple and elegant as the original.

Kozen's axiomatisation was given in equational form. As we prefer working with Hilbert-style axiom systems, an axiom system based on Kozen's formulation is used here. We first briefly describe axiom systems in general.

Axiom system. Generally, an *axiom system* consists of a collection of *inference rules* of the form

$$\mathbf{R} : \frac{\phi_1, ..., \phi_k}{\phi},$$

where $k \geq 0$ and $\phi_1, ..., \phi_k, \phi$ are formula schemata[2]; $\phi_1, ..., \phi_k$ are called the *assumptions* and ϕ is called the *conclusion*. The inference rules with the empty set of assumptions (i.e. $k = 0$), called *axioms*, are usually distinguished from other inference rules.

The set of *theorems* in an axiom system is defined to be the smallest set Λ which contains all the instances of each axiom and is closed under each inference rule, i.e. for each instance of each inference rule, if the assumptions are in the set then so is the conclusion. Equivalently, one may define a theorem to be a formula ϕ for which there exists a finite sequence, called a *derivation*, $\phi_1, ..., \phi_n = \phi$ $(n \geq 1)$, such that each ϕ_i is either an instance of an axiom or is the conclusion of an instance of a rule whose assumptions are among $\phi_1, ..., \phi_{i-1}$. We write $\vdash \phi$ when ϕ is a theorem, and also say that ϕ is *provable* in the axiom system. ϕ is said to be *consistent* iff *not* $\vdash \neg\phi$; ϕ is said to be *inconsistent* otherwise.

Definition 2.32. The axiom system **AX** consists of the following axioms and inference rules:[3]

$$\begin{aligned}
\textbf{Taut}: &\quad \phi, \text{where } \phi \text{ is a propositional tautology.}\\
\textbf{K}: &\quad [a](\phi \to \psi) \to ([a]\phi \to [a]\psi)\\
\textbf{Unfold}_\mu: &\quad \phi(\mu X.\phi(X)) \to \mu X.\phi(X)\\
\textbf{Dual}_{\langle\cdot\rangle}: &\quad \langle a\rangle\phi \leftrightarrow \neg[a]\neg\phi\\
\textbf{Dual}_\nu: &\quad \nu X.\phi(X) \leftrightarrow \neg\mu X.\neg\phi(\neg X)\\
\textbf{MP}: &\quad \frac{\psi, \psi \to \phi}{\phi}\\
\textbf{RN}: &\quad \frac{\phi}{[a]\phi}\\
\textbf{R}\mu: &\quad \frac{\phi(\psi) \to \psi}{\mu X.\phi(X) \to \psi}, \text{where } \phi(X) \text{ is positive in } X.
\end{aligned}$$

For the rest of the thesis, the notion of consistency will be based on **AX**.

It is clear that **AX** is essentially the standard system **K**, which is well-known in modal logic ([Che80], [Gol92], [BdV01]), together with the unfolding axiom $\textbf{Unfold}_\mu$, the induction rule $\textbf{R}\mu$ (also called *Park's induction rule* after David Park, who introduced a rule of this form [Par69]), and the duality axioms for $\langle\cdot\rangle$ and ν operators.

It is not surprising that **AX** is sound because every axiom is valid and every inference rule preserves validity.

Theorem 2.33 (Soundness of **AX**). *For any formula ϕ, $\vdash \phi$ implies $\models \phi$.*

Proof. We only need to check that all instances of each axiom are valid and each

[2]It should be sufficient for our discussion to think of a formula schema as a representation of a set of formulae, called instances, which fit the specified pattern.

[3]In [Koz83], the ν and $[\cdot]$ operators were treated as abbreviations. Hence the duality axioms, like $\textbf{Dual}_{\langle\cdot\rangle}$ and $\textbf{Dual}_\nu$, were not included in the original axiomatisation.

inference rule preserves validity. This is the case for **Taut**, **MP**, **RN**, **K**, and $\mathbf{Dual}_{\langle\cdot\rangle}$, as is known from modal logic. The rest follows from Proposition 2.13. □

We now look at some properties of this axiom system. The following properties are derivable in system **K**, so it is not surprising that they are derivable here.

Proposition 2.34.

(a) $\vdash \phi_1 \wedge ... \wedge \phi_n \to \phi$ *implies* $\vdash [a]\phi_1 \wedge ... \wedge [a]\phi_n \to [a]\phi$, *for all* $n \geq 0$.
(b) $\vdash [a]\phi_1 \wedge ... \wedge [a]\phi_n \leftrightarrow [a](\phi_1 \wedge ... \wedge \phi_n)$, *for all* $n \geq 0$.
(c) $\vdash \phi \to \psi$ *implies* $\vdash \langle a\rangle\phi \to \langle a\rangle\psi$.

Proof. These are well-known properties in system **K**. □

One of the most basic properties of axiom systems is that a *uniform substitution* preserves theoremhood: if ϕ is provable then every formula resulted from uniformly replacing each proposition letter or each free occurrences of a variable by some formula is also provable. This is not surprising, considering the fact that any instance of an axiom or inference rule can be used to derive a theorem.

Proposition 2.35 (Uniform Substitution)**.** *For any formulae* ϕ, γ, *variable* Y, *and proposition letter* P, *if* $\vdash \phi$ *then* $\vdash \phi\{\gamma/Y\}$ *and* $\vdash \phi\{\gamma/P\}$.

Proof. This follows from the fact that

- for each axiom A, if ϕ is an instance of A, then so is $\phi\{\gamma/Y\}$, and
- for each rule R, if ϕ is the consequence of an instance of R whose assumptions are $\psi_1, ..., \psi_n$, then if $\psi_1\{\gamma/Y\}, ..., \psi_n\{\gamma/Y\}$ are provable, then so is $\phi\{\gamma/Y\}$.

□

Proposition 2.36. *Suppose* $\phi(X), \psi(X)$ *are positive in* X.

(a) $\vdash \phi(X) \to \psi(X)$ *implies* $\vdash \mu X.\phi(X) \to \mu X.\psi(X)$.

(b) $\vdash \phi(X) \to \psi(X)$ *implies* $\vdash \nu X.\phi(X) \to \nu X.\psi(X)$.

Proof. (a) Suppose $\vdash \phi(X) \to \psi(X)$. For safety, we first rename each free variable other than X in $\phi(X) \to \psi(X)$ to some new variable. Let ρ be a substitution for such renaming, and let $\phi'(X) = \phi(X)\rho$ and $\psi'(X) = \psi(X)\rho$. It is then safe to substitute $\mu X.\psi'(X)$ for X in $\phi'(X)$ and $\psi'(X)$. Hence

$\vdash \phi'(X) \to \psi'(X)$ (by uniform substitution)
$\vdash \phi'(\mu X.\psi'(X)) \to \psi'(\mu X.\psi'(X))$ (by uniform substitution)
$\vdash \phi'(\mu X.\psi'(X)) \to \mu X.\psi'(X)$ (by $\mathbf{Unfold}_\mu$)
$\vdash \mu X.\phi'(X) \to \mu X.\psi'(X)$ (by $\mathbf{R}\mu$)
$\vdash \mu X.\phi(X) \to \mu X.\psi(X)$ (by uniform substitution)

In the last step, we use the inverse of the substitution ρ to rename the free variables in $\mu X.\phi'(X)$ and $\mu X.\psi'(X)$ back to original.

(b) Suppose $\vdash \phi(X) \to \psi(X)$. Then

$\vdash \phi(\neg X) \to \psi(\neg X)$ (by uniform substitution)
$\vdash \neg\psi(\neg X) \to \neg\phi(\neg X)$
$\vdash \mu X.\neg\psi(\neg X) \to \mu X.\neg\phi(\neg X)$ (by (a))
$\vdash \neg\mu X.\neg\phi(\neg X) \to \neg\mu X.\neg\psi(\neg X)$
$\vdash \nu X.\phi(X) \to \nu X.\psi(X)$ (by $\mathbf{Dual}_\nu$)

□

Here is the axiomatic counterpart of Proposition 2.7.

Proposition 2.37 (Monotonicity)**.** *For any formulae $\phi(X), \psi, \psi'$,*

(a) if $\phi(X)$ is positive in X, then $\vdash \psi \to \psi'$ implies $\vdash \phi(\psi) \to \phi(\psi')$;
(b) if $\phi(X)$ is negative in X, then $\vdash \psi \to \psi'$ implies $\vdash \phi(\psi') \to \phi(\psi)$.

Proof. We prove by induction on $\phi(X)$ that (a) and (b) hold for any formulae ψ, ψ'.

- $\phi(X)$ is a proposition letter or a variable. Obvious.
- $\phi(X) = \neg\phi'(X)$. (a) If $\phi(X)$ is positive in X, then $\phi'(X)$ must be negative in X. By induction, $\vdash \psi \to \psi'$ implies $\vdash \phi'(\psi') \to \phi'(\psi)$. Hence $\vdash \neg\phi'(\psi) \to \neg\phi'(\psi')$. (b) If $\phi(X)$ is negative in X, then $\phi'(X)$ must be positive in X. By induction, $\vdash \psi \to \psi'$ implies $\vdash \phi'(\psi) \to \phi'(\psi')$. Hence $\vdash \neg\phi'(\psi') \to \neg\phi'(\psi)$.
- $\phi(X) = \phi_1(X) \vee \phi_2(X)$. (a) If $\phi(X)$ is positive in X, then so are both $\phi_1(X)$ and $\phi_2(X)$. By induction, $\vdash \psi \to \psi'$ implies $\vdash \phi_1(\psi) \to \phi_1(\psi')$ and $\vdash \phi_2(\psi) \to \phi_2(\psi')$. This implies that $\vdash \phi_1(\psi) \vee \phi_2(\psi) \to \phi_1(\psi') \vee \phi_2(\psi')$. (b) is similar.
- $\phi(X) = \phi_1(X) \wedge \phi_2(X)$. Similar to the previous case.
- $\phi(X) = \langle a\rangle\phi'(X)$. (a) If $\phi(X)$ is positive in X, then so is $\phi'(X)$. By induction, $\vdash \psi \to \psi'$ implies $\vdash \phi'(\psi) \to \phi'(\psi')$. By Proposition 2.34(c), $\vdash \langle a\rangle\phi'(\psi) \to \langle a\rangle\phi'(\psi')$. (b) is similar.
- $\phi(X) = [a]\phi'(X)$. (a) If $\phi(X)$ is positive in X, then so is $\phi'(X)$. By induction, $\vdash \psi \to \psi'$ implies $\vdash \phi'(\psi) \to \phi'(\psi')$. By Proposition 2.34(a), $\vdash [a]\phi'(\psi) \to [a]\phi'(\psi')$. (b) is similar.
- $\phi(X) = \sigma Y.\phi'(Y, X)$. If $Y = X$, then X does *not* occur free in $\phi(X)$, and hence the induction hypothesis trivially holds. Suppose $Y \neq X$. For (a), by induction, $\vdash \psi \to \psi'$ implies $\vdash \phi'(Y, \psi) \to \phi'(Y, \psi')$. By Proposition 2.36, $\vdash \sigma Y.\phi'(Y, \psi) \to \sigma Y.\phi'(Y, \psi')$. (b) is similar.

□

The following properties are quite straightforward to prove.

Proposition 2.38. *Suppose $\phi(X), \psi$ are any formulae where $\phi(X)$ is positive in X.*

(a) $\vdash \nu X.\phi(X) \to \phi(\nu X.\phi(X))$.
(b) $\vdash \psi \to \phi(\psi)$ implies $\vdash \psi \to \nu X.\phi(X)$.
(c) $\vdash \phi(\sigma X.\phi(X)) \leftrightarrow \sigma X.\phi(X)$.
(d) $\vdash \phi \leftrightarrow \sigma X.\phi$ if ϕ has no free occurrences of X.

(e) $\vdash \sigma X.\phi(X) \leftrightarrow \sigma Y.\phi(Y)$*, provided that* Y *does* not *occur free in* $\phi(X)$*.*

(f) $\vdash \phi(\mu X.\psi \wedge \phi(X)) \rightarrow \psi$ *implies* $\vdash \mu X.\phi(X) \rightarrow \psi$*, provided that* X *does not occur free in* ψ*.*

Proof. (a) By $\mathbf{Unfold}_\mu$, $\vdash \neg\phi(\neg\mu X.\neg\phi(\neg X)) \rightarrow \mu X.\neg\phi(\neg X)$. Hence, $\vdash \neg\mu X.\neg\phi(\neg X) \rightarrow \phi(\neg\mu X.\neg\phi(\neg X))$.

From $\mathbf{Dual}_\nu$, $\nu X.\phi(X) \leftrightarrow \neg\mu X.\neg\phi(\neg X)$. By monotonicity (Proposition 2.37(a)), we may infer that $\vdash \phi(\nu X.\phi(X)) \leftrightarrow \phi(\neg\mu X.\neg\phi(\neg X))$ and therefore $\vdash \nu X.\phi(X) \rightarrow \phi(\nu X.\phi(X))$.

(b) Suppose $\vdash \psi \rightarrow \phi(\psi)$. Then

$\vdash \neg\phi(\psi) \rightarrow \neg\psi$
$\vdash \neg\phi(\neg\neg\psi) \rightarrow \neg\psi$ (by monotonicity)
$\vdash \mu X.\neg\phi(\neg X) \rightarrow \neg\psi$ (by $\mathbf{R}\mu$)
$\vdash \psi \rightarrow \neg\mu X.\neg\phi(\neg X)$
$\vdash \psi \rightarrow \nu X.\phi(X)$ (by $\mathbf{Dual}_\nu$)

(c) By $\mathbf{Unfold}_\mu$, $\vdash \phi(\mu X.\phi(X)) \rightarrow \mu X.\phi(X)$.

$\vdash \phi(\phi(\mu X.\phi(X))) \rightarrow \phi(\mu X.\phi(X))$ (by monotonicity)
$\vdash \mu X.\phi(X) \rightarrow \phi(\mu X.\phi(X))$ (by $\mathbf{R}\mu$)

Together with $\mathbf{Unfold}_\mu$, we have $\vdash \phi(\mu X.\phi(X)) \leftrightarrow \mu X.\phi(X)$. The proof for $\sigma = \nu$ is similar (using (a) and (b) instead of $\mathbf{Unfold}_\mu$ and $\mathbf{R}\mu$).

(d) Follows immediately from (c).

(e) We assume that $\sigma Y.\phi(Y)$ is safe for X in $\phi(X)$ (otherwise, apply a uniform substitution to rename the free variables in $\sigma Y.\phi(Y)$ to some new variables, and after proceeding as below, rename the variables back to original). From $\mathbf{Unfold}_\mu$, $\vdash \phi(\mu Y.\phi(Y)) \rightarrow \mu Y.\phi(Y)$. Applying $\mathbf{R}\mu$, we obtain $\vdash \mu X.\phi(X) \rightarrow \mu Y.\phi(Y)$. For $\sigma = \nu$, by (a), $\vdash \nu Y.\phi(Y) \rightarrow \phi(\nu Y.\phi(Y))$. Applying (b), we obtain $\vdash \nu Y.\phi(Y) \rightarrow \nu X.\phi(X)$. The other direction can be shown similarly.

(f) Suppose $\vdash \phi(\mu X.\psi \wedge \phi(X)) \rightarrow \psi$.

$\vdash \phi(\mu X.\psi \wedge \phi(X)) \rightarrow \psi \wedge \phi(\mu X.\psi \wedge \phi(X))$
$\vdash \phi(\mu X.\psi \wedge \phi(X)) \rightarrow \mu X.\psi \wedge \phi(X)$ (by $\mathbf{Unfold}_\mu$)
$\vdash \mu X.\phi(X) \rightarrow \mu X.\psi \wedge \phi(X)$ (by $\mathbf{R}\mu$)
$\vdash \mu X.\phi(X) \rightarrow \psi$ (by (c))

□

A rule is said to be *admissible* (in an axiom system) iff, for any instance of the rule, the conclusion of the rule is provable assuming that the assumptions of the rule are added as extra axioms. The following rules are admissible in **AX** (see Proposition 2.34,

2.37, 2.38).

$$\begin{aligned}
\mathbf{RK}: &\quad \frac{\phi_1 \wedge ... \wedge \phi_n \to \phi}{[a]\phi_1 \wedge ... \wedge [a]\phi_n \to [a]\phi}, \quad n \geq 0 \\
\mathbf{Unfold}_\nu: &\quad \phi \to \phi(\nu X.\phi(X)) \\
\mathbf{R}\nu: &\quad \frac{\psi \to \phi(\psi)}{\psi \to \nu X.\phi(X)}, \qquad \phi(X) \text{ is positive in } X. \\
\mathbf{Mon}: &\quad \frac{\psi \to \psi'}{\phi(\psi) \to \phi(\psi')}, \qquad \phi(X) \text{ is positive in } X.
\end{aligned}$$

The following is some less trivial properties. The first three are taken from [AN01]. The last one, as far as we know, has not been mentioned before. The derivations for these formulae, though not necessarily long, require some ingenuity.

Proposition 2.39. *Suppose $\phi(X,Y), \psi(X), \gamma$ are any formulae.*

(a) $\vdash \sigma X.\sigma Y.\phi(X,Y) \leftrightarrow \sigma X.\phi(X,X)$.
(b) $\vdash \sigma X.\sigma Y.\phi(X,Y) \leftrightarrow \sigma Y.\sigma X.\phi(X,Y)$.
(c) $\vdash \mu X.\nu Y.\phi(X,Y) \to \nu Y.\mu X.\phi(X,Y)$.
(d) $\vdash \sigma X.\phi(\psi(X)) \leftrightarrow \phi(\sigma X.\psi(\phi(X)))$*, for any* $\phi(X), \psi(X)$ positive *in* X.

Proof. For (a) - (c), see [AN01].

For (d), suppose $\sigma = \mu$.

$\vdash \psi(\phi(\mu X.\psi(\phi(X)))) \to \mu X.\psi(\phi(X))$ (by $\mathbf{Unfold}_\mu$)
$\vdash \phi(\psi(\phi(\mu X.\psi(\phi(X))))) \to \phi(\mu X.\psi(\phi(X)))$ (by **Mon**)
$\vdash \mu X.\phi(\psi(X)) \to \phi(\mu X.\psi(\phi(X)))$ (by $\mathbf{R}\mu$)

For the other direction,

$\vdash \phi(\psi(\mu X.\phi(\psi(X)))) \to \mu X.\phi(\psi(X))$ (by $\mathbf{Unfold}_\mu$)
$\vdash \psi(\phi(\psi(\mu X.\phi(\psi(X))))) \to \psi(\mu X.\phi(\psi(X)))$ (by **Mon**)
$\vdash \mu X.\psi(\phi(X)) \to \psi(\mu X.\phi(\psi(X)))$ (by $\mathbf{R}\mu$)
$\vdash \phi(\mu X.\psi(\phi(X))) \to \phi(\psi(\mu X.\phi(\psi(X))))$ (by **Mon**)
$\vdash \phi(\mu X.\psi(\phi(X))) \to \mu X.\phi(\psi(X))$ (by $\mathbf{Unfold}_\mu$)

The case where $\sigma = \nu$ can be shown similarly. □

The following equivalence results can be shown similarly to their semantic counterpart.

Proposition 2.40. *Every closed formula is provably equivalent (in* $\boldsymbol{AX}$*) to a formula in positive normal form.*

Proof. All the rules in Proposition 2.14 are justified in **AX**. □

Proposition 2.41. *Every formula is provably equivalent (in* $\boldsymbol{AX}$*) to a guarded formula.*

Proof. As for Proposition 2.18. □

Chapter 3

Pre-Models and Tableaux

The semantics of the modal μ-calculus does not lend itself to efficient computation. To determine from the definition whether a formula is true at a state, one needs to compute the set of states at which the formula holds and then check that the formula is in the set. This problem sparked the research in model checking for the modal μ-calculus in the late 90s. One of the first results is the tableau systems by Larsen in [Lar90]. In the paper, two fixpoint extensions of Hennessy-Milner logic are considered, one in which the fixpoint is interpreted as the greatest one and another as the least one. A tableau system for model checking formulae over finite models is proposed for each of these extensions. As in the tableau method for modal logics, Larsen's tableau systems proceed by structural induction on the given formula. For example, to show that $s \models \psi_1 \wedge \psi_2$ ($s \models \psi_1 \vee \psi_2$) we need to show that $s \models \psi_1$ and $s \models \psi_2$ ($s \models \psi_1$ or $s \models \psi_2$); and to show that $s \models \langle a \rangle \psi$ ($s \models [a]\psi$) we need to show that $s' \models \psi$ for some (for all) a-successor state s'. Proof rules corresponding to these facts are introduced into the tableau system. As clearly seen, this method is more *local* in the sense that only certain states required in establishing the goal are considered. For the modal μ-calculus, the key is how to deal with the fixpoint formulae. From a basic property of fixpoints, we know that $s \models \sigma X.\psi$ if $s \models \psi\{\sigma X.\psi/X\}$. Since the latter formula may contain the original fixpoint formula, this unfolding rule, if added to system, may lead to a repeating configuration and hence non-termination. This is where the distinction between least fixpoints and greatest fixpoints are made in [Lar90]: repetition is good in the greatest-fixpoint case, but bad in the least-fixpoint one. To explain the correctness of these termination rules precisely would require some extra notation. But the underlying idea is quite simple and is based on the following facts:

(1) $s \in \|\nu X.\psi\|$ if, for some $S \subseteq \|\nu X.\psi\|$, $s \in \|\psi\|(S \cup \{s\})$.
(2) If $s \in \|\mu Z.\psi\|$ then, for some $S \subseteq \|\mu Z.\psi\|$, $s \in \|\psi\|(S - \{s\})$.

From (1), to show that $s \models \nu X.\psi$ we may proceed by showing that $s \models \psi\{\nu X.\psi/X\}$ using the assumption that $s \models \nu X.\psi$; this explains why the tableau proof may (successfully) terminate if it reaches $s \models \nu X.\psi$ again. On other hand, (2) suggests that if

$\mu Z.\psi$ is actually true at s there must be a tableau proof from $s \models \psi\{\mu Z.\psi/Z\}$ which never reaches $s \models \mu Z.\psi$ again, i.e. a repetition of the latter can always be avoided.

Based on Larsen's tableau method, Stirling and Walker [SW91] later proposed a tableau system for the full modal μ-calculus, not just a logic which is limited to one kind of fixpoints. The termination rules are based on the idea described above. However, to keep track of the regeneration of different fixpoint formulae, *constants* were introduced. So a repetition of the goal U at state s is allowed if and only if U is a constant for a ν-formula. Bradfield and Stirling later extended this tableau system for infinite-state systems.

Streett and Emerson in [SE89] introduced the notion of well-founded pre-models. A pre-model is simply a structure which is annotated by formulae such that each annotation is *locally consistent*. Within a pre-model, there are regeneration paths of formulae through the states in the pre-model. A main theorem in the paper states that a pre-model will be a model for the annotating formulae if there are no "bad" regeneration paths – the paths in which some μ-variable regenerates itself infinitely often. Such a pre-model is said to be *well-founded*. The converse of this theorem is also true: every satisfiable formula has a well-founded pre-model. These theorems entail many results about the modal μ-calculus. For model checking, to show that a formula is true at a state in a model, one tries annotating its subformulae over the states in the model to form a well-founded pre-model. More importantly, the theorems can be used to solve the satisfiability problem. To show that a formula is satisfiable, one tries to find a well-founded pre-model for it. This is where the automata-theoretic technique was employed in [SE89]. From the given formula, a *Streett tree automaton* [Str81] recognising all the tree pre-models which are well-founded can be constructed. Thus the formula is satisfiable if and only if such an automaton accepts a *non-empty* language. The latter is known to be decidable from automata theory. Moreover, it is known that a Streett tree automaton which accepts a non-empty language must accept a tree obtained from unwinding a finite graph (of bounded size). This latter finite graph is a model of the formula; hence the *small model property* is obtained.

One last contribution of [SE89] is the notion of *signatures*. Signatures generalise the notion of approximants for a single fixpoint formula to a formula with nested fixpoints. Hence, instead of being a single ordinal, a signature is a sequence of ordinals each of which is interpreted as an approximant for a μ-subformula. Signatures were used in [SE89] as a measure in showing the existence of a well-founded regeneration path on a model. Since its first appearance, signatures have become an invaluable tool for the modal μ-calculus.

The rest of this chapter is organised as follows. The first part concerns the notions of pre-models, trails, and signatures, and the proofs of the main theorems in [SE89]. In the second part, a tableau system for establishing satisfiability (called TS_0 in this thesis) is given. The tableau system is similar to the one in [NW97] and can be seen as another representation of pre-models.

Note that although most of the terminology used here are from the original paper [SE89], the notation and definitions given below follow more closely [BS07] and [Sti00].

Convention. For the rest of this chapter, all formulae are assumed to be in positive normal form.

3.1 Pre-Models

For the rest of this section, fix a closed formula ϕ in positive normal form. We could relax the closedness assumption on ϕ with some minor modification to the definitions below. We opt *not* to do so to avoid unnecessary confusion (obviously the free variables in ϕ could be renamed to some propositional letters without changing the satisfiability of the formula).

Definition 3.1 (Annotated Structures)**.** Let $\mathcal{S} = \langle S, \{R_a\}_{a \in \mathrm{Act}} \rangle$ be a transition system. An *annotation of the subformulae of ϕ on $\mathcal{S}$* is any function $\Delta : S \to \wp\mathrm{Sub}(\phi)$. An annotation Δ is said to be *locally consistent* on $\mathcal{S}$ iff the following conditions are satisfied for all states s:

LC1. $\Delta(s)$ does *not* contain a complementary pair of literals,
LC2. $\psi_1 \vee \psi_2 \in \Delta(s)$ implies $\psi_i \in \Delta(s)$ for some $i \in \{1, 2\}$,
LC3. $\psi_1 \wedge \psi_2 \in \Delta(s)$ implies $\psi_i \in \Delta(s)$ for each $i \in \{1, 2\}$,
LC4. $\langle a \rangle \psi \in \Delta(s)$ implies $\psi \in \Delta(t)$, for some $t \in R_a(s)$,
LC5. $[a]\psi \in \Delta(s)$ implies $\psi \in \Delta(t)$, for each $t \in R_a(s)$,
LC6. $\sigma X.\psi \in \Delta(s)$ implies $X \in \Delta(s)$,
LC7. $X \in \Delta(s)$ implies $\psi \in \Delta(s)$, where X identifies $\sigma X.\psi$.

A *locally-consistent annotated structure* is a pair $\langle \mathcal{S}, \Delta \rangle$, where $\mathcal{S}$ is a transition system and Δ is a *locally-consistent* annotation (of the subformulae of some formula ϕ) on $\mathcal{S}$.

Given a locally-consistent annotated structure, it is straightforward to define a model from it. Precisely, a *model based on an annotated structure* $\langle \mathcal{S}, \Delta \rangle$ is any model $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}} \rangle$ such that, for each proposition letter P and state s,

- $P \in \Delta(s)$ implies $s \in \mathcal{V}_{\mathrm{Prop}}(P)$, and
- $\neg P \in \Delta(s)$ implies $s \notin \mathcal{V}_{\mathrm{Prop}}(P)$.

Clearly, the condition LC1 ensures that every locally-consistent annotated structure has a model based on it.

In modal logic, we know that if there is a locally-consistent annotation on $\mathcal{S}$, then any model based on such annotated structure satisfies the annotating formulae. This is however *not* true for the modal μ-calculus; particularly when there are least-fixpoint formulae annotating some state. For example, suppose $\mathcal{S}$ contains one state s annotated by $\{Z, \mu Z.Z\}$. Clearly this annotation is locally consistent yet obviously $\mu Z.Z$, which is equivalent to $\bot$, is not true in any model. Later we will show that if the annotated

structure satisfies certain "global condition", then it can be seen as a model for the annotating formulae.

Given a model $\mathcal{M}$, if we annotate each state in $\mathcal{M}$ by the subformulae of ϕ true at that state, we clearly obtain a locally-consistent structure. This can be stated formally as follows. First, since a subformula of ϕ may contain free occurrences of variables, following [Sti00], we define the valuation which assigns their intended meanings to those variables. To do so, we assume a sequence $\sigma_1 X_1.\psi_1, ..., \sigma_n X_n.\psi_n$ of all the fixpoint subformulae of ϕ such that X_i higher than X_j implies $i < j$.

Definition 3.2. The *valuation $\mathcal{V}_{\mathcal{M},\phi}$ for the fixpoint variables of ϕ on $\mathcal{M}$* is defined to be $\mathcal{V}_n$, where $\mathcal{V}_0, ..., \mathcal{V}_n$ are defined inductively as follows:

- $\mathcal{V}_0(X) = \emptyset$ for all variables X;
- $\mathcal{V}_{i+1} = \mathcal{V}_i[X_{i+1} := \|\sigma_{i+1} X_{i+1}.\psi_{i+1}\|_{\mathcal{V}_i}]$.

Although the definition of $\mathcal{V}_{\mathcal{M},\phi}$ is given based on the above sequence of fixpoint formulae, it is clear that any sequence of fixpoint formulae (in decreasing length) will produce the same valuation. We omit the subscript $\mathcal{M}, \phi$ when there is no ambiguity. Morever, from now on, we simply write $\mathcal{M}, s \models \psi$ for $\mathcal{M}, s \models_{\mathcal{V}_{\mathcal{M},\phi}} \psi$.

Definition 3.3. The *canonical annotation of the subformulae of ϕ on $\mathcal{M}$* is the function $\Delta_{\mathcal{M},\phi} : S \to \wp\mathrm{Sub}(\phi)$ where

$$\Delta_{\mathcal{M},\phi}(s) \quad = \quad \{\psi \in \mathrm{Sub}(\phi) \,|\, \mathcal{M}, s \models \psi\}.$$

Proposition 3.4. *For any model $\mathcal{M}$, the canonical annotation $\Delta_{\mathcal{M},\phi}$ is locally consistent.*

Proof. This follows directly from the semantics. □

We call the structure $\langle \mathcal{M}, \Delta_{\mathcal{M},\phi} \rangle$ (for any formula ϕ) an *annotated model*.

As mentioned, the existence of a locally-consistent annotation on a transition system does not generally imply that the models based on such annotated structure satisfy the formulae annotating each state. To understand why, we need to see that the local-consistency conditions LC1-LC7 are meant to capture the *dependency* of the truth of a formula on the truth of the constituent subformulae at nearby states. For example, the truth of $\psi_1 \vee \psi_2$ at state s depends on the truth of some formula ψ_i at s, the truth of $\langle a \rangle \psi$ at s depends on the truth of ψ at some state $t \in R_a(s)$, etc. The interesting case is the truth of a variable X at s which depends on the truth of its unfolding ψ at s. Without fixpoint formulae, every chain of dependencies will eventually lead to a literal (e.g. $P, \neg P$) at some state. Clearly, by definition, such literal is true at that state in any model based on the annotated structure. But if there is a fixpoint formula, there might be an infinite chain of dependencies. This explains why some formula might not be true in the corresponding models. However, if we can show that there is no infinite chain of dependencies where a μ-variable is unfolded infinitely often, then it can be

guaranteed that the annotated formulae are true in any model based on the annotated structure. This result should not be surprising if we realise the fact that if a μ-formula $\mu Z.\psi$ is true at state s, then some approximant $\mu^{\alpha} Z.\psi$ must be true at s. And from the semantics, $\mu^{\alpha} Z.\psi$ is true at s if $\psi\{\mu^{\alpha'} Z.\psi\}$, for some ordinal $\alpha' < \alpha$, is true at s. This means that if $\mu Z.\psi$ is actually true at s, the chain of dependencies from $\mu Z.\psi$ at s will eventually terminate. To state this result formally, we need some terminology.

Definition 3.5 (Dependency Relations)**.** Suppose $\langle \mathcal{S}, \Delta \rangle$ is a locally-consistent annotated structure. A *dependency relation*, typically denoted by $\rightarrow$ in the thesis, on $\langle \mathcal{S}, \Delta \rangle$ is a relation over the set of pairs (s, ψ), where ψ annotates s, such that

DR1. $(s, \psi_1 \vee \psi_2) \rightarrow (s, \psi_i)$ for some $i \in \{1, 2\}$,
DR2. $(s, \psi_1 \wedge \psi_2) \rightarrow (s, \psi_i)$ for each $i \in \{1, 2\}$,
DR3. $(s, \langle a \rangle \psi) \rightarrow (t, \psi)$ for some $t \in R_a(s)$,
DR4. $(s, [a]\psi) \rightarrow (t, \psi)$ for each $t \in R_a(s)$,
DR5. $(s, \sigma X.\psi) \rightarrow (s, X)$,
DR6. $(s, X) \rightarrow (s, \psi)$ where X identifies $\sigma X.\psi$.

Note that, in [SE89] and [BS07], dependency relations (called *derivation relations* in [SE89]) are *determined* from a *choice function*. A choice function is a function which for every pair $(s, \psi_1 \vee \psi_2)$ chooses one disjunct ψ_i and for every pair $(s, \langle a \rangle \psi)$ chooses one state $t \in R_a(s)$ annotated with ψ. The dependency relation $\rightarrow$ determined by the choice function f satisfies the above conditions except that the first and the third condition are replaced with

- $(s, \psi_1 \vee \psi_2) \rightarrow (s, f(s, \psi_1 \vee \psi_2))$,
- $(s, \langle a \rangle \psi) \rightarrow (f(s, \langle a \rangle \psi), \psi)$.

As we do not need choice functions in our thesis, we decide to define dependency relations as a first-class object. Our definition of dependency relations is slighly more generous than the one in [SE89] and [BS07] in that it allows $(s, \psi_1 \vee \psi_2) \rightarrow (s, \psi_i)$ for more than one i and $(s, \langle a \rangle \psi) \rightarrow (t, \psi)$ for more than one $t \in R_a(s)$. The difference is, however, very superficial.

Definition 3.6 (Pre-Models)**.** A *pre-model* is a triple $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ where $\langle \mathcal{S}, \Delta \rangle$ is a locally-consistent annotated structure and $\rightarrow$ is a dependency relation on $\langle \mathcal{S}, \Gamma \rangle$.

Definition 3.7 (Trails)**.** Suppose $\mathcal{P} = \langle \mathcal{S}, \Gamma, \rightarrow \rangle$ is a pre-model. A trail in pre-model $\mathcal{P}$ is a path over the dependency relation $\rightarrow$. Thus a trail can be written as:

$$(s_1, \psi_1) \rightarrow ... \rightarrow (s_n, \psi_n) \rightarrow ...$$

We first describe some terminology for trails used in the thesis. Most of these are obvious. A trail with (s, ϕ) as the first element is called a *trail from* (s, ϕ), or more generally, a *trail from state* s. Similarly, a *finite* trail with (s, ϕ) as the last element

is called a *trail to* (s, ϕ) or a *trail to state* s. A *subtrail* of trail τ is a subsequence of *consecutive* pairs in τ.

Suppose $\tau = (s_1, \psi_1) \rightarrow (s_2, \psi_2) \rightarrow ...$ is a trail, and suppose $\psi_{i_1}, \psi_{i_2}, ...$ are the sequence of all the occurrences of modal formulae (e.g. $\langle a \rangle \psi$ or $[a]\psi$) along this trail. Hence, by definition, $s_{i_j} R_{a_j} s_{i_{j+1}}$, for each $j \geq 1$ and some action a_j. Thus $s_{i_1} R_{a_1} s_{i_2} R_{a_2} ...$ is a path in $\mathcal{S}$. We say that τ is a *trail along* this path.

A variable X is said to be *active* in a trail τ iff X is active in each formula in τ. An *unfolding of* X *in trail* τ is a subtrail of τ of the form $(s, X) \rightarrow (s, \psi)$, for some state s. X is said to *unfold* in τ iff τ contains an occurrence of an unfolding of X; it is said to *unfold infinitely often* iff τ contains infinitely many occurrences of unfoldings of X.

Note that the notion of *trails* appears in the literature under different names, e.g. *derivation sequences* in [SE89] or *traces* in [NW97], [Wal93]. We adopt the term *trail* from [BS92], [BS07].

One simple fact about trails is that, among the variables unfolded infinitely often on an infinite trail, there is a (unique) outermost variable. Hence infinite trails may be classified into two types.

Definition 3.8 (μ-Trails and ν-Trails)**.** A μ-trail (ν-trail) is an infinite trail in which the outermost variable unfolded infinitely often is a μ-variable (respectively, ν-variable).

Definition 3.9 (Well-Founded Pre-Models)**.** A pre-model is said to be *well-founded* iff every infinite trail in the pre-model is a ν-trail.

One simple observation which we frequently use is that every μ-trail (ν-trail) must contain a suffix in which some μ-variable (respectively, ν-variable) is active throughout and unfolded infinitely often. Hence a pre-model is well-founded iff there is no trail along which some μ-variable is active and unfolded infinitely often.

The main results of this chapter can now be stated.

- For any well-founded pre-model $\mathcal{P}$ and any model $\mathcal{M}$ based on it, the formulae annotating each state are true at that state in $\mathcal{M}$.
- Conversely, for every annotated model, there exists a dependency relation which makes the structure a well-founded pre-model.

These two statements, which were first stated and proved in [SE89] (under a slightly different definition of pre-models), are of significant importance to the understanding of the modal μ-calculus. Of particular interest is the first statement which describes the condition for which an annotated structure is a model for the annotating formulae. As it is so fundamental to the modal μ-calculus, we follow [BS07] and call it the *Fundamental Semantic Theorem of the Modal* μ*-Calculus*.

Not only are these results interesting in themselves, their proofs in [SE89] utilise a new tool called *signatures* which turns out to be very useful when proving properties of the logic. In what follows, we shall explain signatures and give a proof of the above

statements. Instead of using the original definition in [SE89], we follow the definition of signatures in [Sti00], as we find it more intuitive and easier to use.

3.2 Signatures

Let ϕ be a closed formula in positive normal form. A signature associates an ordinal to each μ-variable in ϕ. Since signatures are to be used as a well-ordered measure, a signature is defined to be a sequence of ordinals corresponding to some fixed sequence of the μ-variables in ϕ.

Definition 3.10 (Signatures)**.** Fix a sequence $X_1, ..., X_m$ of all the variables in ϕ such that X_i higher than X_j implies $i < j$, and the subsequence $Z_1, ..., Z_n$ of the μ-variables. A *signature* is a sequence $\langle \alpha_1, ..., \alpha_n \rangle$ of ordinals.

Definition 3.11 (Ordering of Signatures)**.** Signatures are *well ordered* lexicographically: $\langle \alpha_1, ..., \alpha_n \rangle < \langle \alpha'_1, ..., \alpha'_n \rangle$ iff $\alpha_j < \alpha'_j$ and $\alpha_i = \alpha'_i$ for some j and each $i < j$. As usual, $\sigma \leq \sigma'$ iff $\sigma < \sigma'$ or $\sigma = \sigma'$.

Remark 3.12. For brevity, we adopt some shorthand notations for signatures. Suppose $\sigma = \langle \alpha_1, ..., \alpha_i, ..., \alpha_n \rangle$ is a signature. For each μ-variable Z_i,

- $\sigma(Z_i) = \alpha_i$,
- $\sigma[Z_i := \beta] = \langle \alpha_1, ..., \beta, ..., \alpha_n \rangle$,
- $\sigma \restriction Z_i = \sigma \restriction i = \langle \alpha_1, ..., \alpha_i \rangle$. Similarly for a ν-variable X_j, $\sigma \restriction X_j = \langle \alpha_1, ..., \alpha_i \rangle$ if Z_i is the last μ-variable preceding X_j in the sequence $X_1, ..., X_m$.

Definition 3.13 (Signature Assignments)**.** Suppose $\langle \mathcal{S}, \Delta \rangle$ is a locally-consistent annotated structure. A *signature assignment* on $\langle \mathcal{S}, \Delta \rangle$ is a function sig assigning a signature to each pair (s, ψ) where $\psi \in \Delta(s)$. A *signature assignment* is said to be *consistent* iff it satisfies the following conditions:

SA1. $\text{sig}(s, \psi_1 \vee \psi_2) \geq \text{sig}(s, \psi_i)$ for some $i \in \{1, 2\}$,
SA2. $\text{sig}(s, \psi_1 \wedge \psi_2) \geq \text{sig}(s, \psi_i)$ for each $i \in \{1, 2\}$,
SA3. $\text{sig}(s, \langle a \rangle \psi) \geq \text{sig}(t, \psi)$ for some $t \in R_a(s)$,
SA4. $\text{sig}(s, [a]\psi) \geq \text{sig}(t, \psi)$ for each $t \in R_a(s)$,
SA5. $\text{sig}(s, \mu Z_j.\psi) \restriction i \geq \text{sig}(s, Z_j) \restriction i$ for each $i < j$,
SA6. $\text{sig}(s, Z_j) \restriction j > \text{sig}(s, \psi) \restriction j$, where Z_j identifies $\mu Z_j.\psi$,
SA7. $\text{sig}(s, \nu X.\psi) \geq \text{sig}(s, X)$,
SA8. $\text{sig}(s, X) \geq \text{sig}(s, \psi)$, where X identifies $\nu X.\psi$.

The existence of a consistent signature assignment on a locally-consistent annotated structure has two consequences. First, it implies that there exists a dependency relation which makes the structure a well-founded pre-model. The converse is also true: every well-founded pre-model has a consistent signature assignment. Hence, consistent signature assignments and well-founded dependency relations (by which we mean the

dependency relation in a well-founded pre-model) are really two sides of the same coin. Secondly, it implies that the formulae annotating each state are true at that state. From these results, the main theorems described earlier easily follow. We first explain the proofs of these two properties of consistent signature assignments.

Proposition 3.14. *Suppose $\langle \mathcal{S}, \Delta \rangle$ is a locally-consistent annotated structure. If there exists a consistent signature assignment for $\langle \mathcal{S}, \Delta \rangle$, then there exists a dependency relation $\rightarrow$ such that $\langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a well-founded pre-model.*

Proof. Suppose sig is a consistent signature assignment on $\langle \mathcal{S}, \Delta \rangle$. Define $\rightarrow$ to be the smallest relation over the domain of pairs (s, ψ) where $\psi \in \Delta(s)$ such that:

- $(s, \psi_1 \vee \psi_2) \rightarrow (s, \psi_i)$ if $\mathrm{sig}(s, \psi_i)$ is the least signature in $\{\mathrm{sig}(s, \psi_1), \mathrm{sig}(s, \psi_2)\}$,
- $(s, \psi_1 \wedge \psi_2) \rightarrow (s, \psi_i)$ for each $i \in \{1, 2\}$,
- $(s, \langle a \rangle \psi) \rightarrow (t, \psi)$ if $\mathrm{sig}(t, \psi)$ is the least signature in $\{\mathrm{sig}(t, \psi) \mid t \in R_a(s)\}$,
- $(s, [a]\psi) \rightarrow (t, \psi)$ for all $t \in R_a(s)$,
- $(s, \sigma X.\psi) \rightarrow (s, X)$,
- $(s, X) \rightarrow (s, \psi)$ if X identifies $\sigma X.\psi$.

By the local-consistency of $\langle \mathcal{S}, \Delta \rangle$, the relation $\rightarrow$ defined above exists and is a dependency relation on the structure. Let $\mathcal{P}$ denote the pre-model $\langle \mathcal{S}, \Delta, \rightarrow \rangle$. It can be shown that $\mathcal{P}$ is a well-founded pre-model. We first consider some property of this dependency relation.

Lemma 3.15. *For any μ-variable Z, if there is a trail in $\langle \mathcal{S}, \Delta, \rightarrow \rangle$ from (s, ψ) to (s', ψ') along which Z is active, then $\mathrm{sig}(s, \psi) \restriction Z \geq \mathrm{sig}(s', \psi') \restriction Z$. Additionally, if Z is unfolded on the trail, then $\mathrm{sig}(s, \psi) \restriction Z > \mathrm{sig}(s', \psi') \restriction Z$.*

Proof. Let $(s_1, \psi_1) \rightarrow \ldots \rightarrow (s_n, \psi_n)$ $(n > 1)$ be any trail from (s, ψ) to (s', ψ') along which Z is active. For each $i \leq n$, it follows from conditions SA1 - SA8 that $\mathrm{sig}(s_i, \psi_i) \restriction Z \geq \mathrm{sig}(s_{i+1}, \psi_{i+1}) \restriction Z$. If $\psi_i = Z$ by condition SA6, $\mathrm{sig}(s_i, Z) \restriction Z > \mathrm{sig}(s_{i+1}, \psi_{i+1}) \restriction Z$. This implies that $\mathrm{sig}(s, \psi) \restriction Z \geq \mathrm{sig}(s', \psi') \restriction Z$ and if Z is unfolded on the trail, then $\mathrm{sig}(s, \psi) \restriction Z > \mathrm{sig}(s', \psi') \restriction Z$. □

It is quite clear from the above lemma that there is no μ-trail in $\mathcal{P}$. Assume otherwise. A μ-trail must contain an infinite subtrail $(s_1, \psi_1) \rightarrow (s_2, \psi_2) \rightarrow \ldots$ along which some μ-variable Z is active and unfolded infinitely often. This means that there is a sequence $(t_1, Z), (t_2, Z), \ldots$ of pairs along this trail. By the previous lemma, the sequence $\mathrm{sig}(t_1, Z) \restriction Z, \mathrm{sig}(t_2, Z) \restriction Z, \ldots$ is an infinite decreasing sequence. This contradicts the fact that the lexicographical ordering of signatures is a well ordering. Hence, $\mathcal{P}$ is a well-founded pre-model. □

We now consider the converse of the above proposition. Given any well-founded pre-model $\mathcal{P}$, there is a natural way to define a consistent signature assignment on $\mathcal{P}$ [SE89]. Basically, we define a signature assignment where, for each μ-variable Z,

the Z-component of the signature for each pair (s, ψ) is determined from the greatest number of unfoldings of Z on any trail from (s, ψ) where Z is active. This is given precisely in Definition 3.17 and 3.19 below.

Proposition 3.16. *Every well-founded pre-model has a consistent signature assignment.*

Proof. Suppose $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a well-founded pre-model.

Definition 3.17. Define the set $|Z^\alpha|$, where Z is a μ-variable and α is an ordinal, by transfinite induction:

- $|Z^0| = \emptyset$.
- $|Z^{\alpha+1}|$ contains all states s annotated by Z such that for every trail from (s, Z) to (t, Z) along which Z is active, $t \in |Z^\alpha|$.
- $|Z^\lambda| = \bigcup_{\alpha<\lambda} |Z^\alpha|$, where λ is a limit ordinal.

It is quite clear that $\alpha \leq \beta$ implies $|Z^\alpha| \subseteq |Z^\beta|$.

Lemma 3.18. *For each μ-variable Z and state s annotated by Z, $s \in |Z^\alpha|$ for some ordinal α.*

Proof. Consider the set S of all states s annotated by Z such that $s \notin |Z^\alpha|$ for any ordinal α. Hence we need to show that S is empty. Suppose otherwise, and let s be a state in S. We can show that, for some $t \in S$, there exists a trail from (s, Z) to (t, Z) along which Z is active. If this is *not* the case, then for any trail from (s, Z) to (t, Z) along which Z is active, $t \in |Z^\alpha|$ for some α. Let β denotes the l.u.b. of all such ordinals α. Thus, for every trail from (s, Z) to (t, Z) along which Z is active, $t \in |Z^\beta|$. But this means that $s \in |Z^{\beta+1}|$ contradicting the assumption that $s \in S$.

Hence, if S is *not* empty, it will be possible to construct an infinite trail along which Z is active and unfolded infinitely often. Since the pre-model is assumed well-founded, it follows that S must be empty. □

Definition 3.19. For each state s annotated by Z, let $\text{rank}(s, Z)$ be the least ordinal α such that $s \in |Z^\alpha|$ (clearly $\text{rank}(s, Z)$ must be a successor ordinal). Define $\text{sig}(s, \psi)$ to be the signature $\langle \alpha_1, ..., \alpha_n \rangle$ where

- $\alpha_i = \bigvee\{\text{rank}(t, Z_i) \mid$ either $(s, \psi) = (t, Z_i)$ or there is a trail from (s, ψ) to (t, Z_i) along which Z_i is active $\}$.

One simple observation is that if Z is *not* active in ψ, $\text{sig}(s, \psi)(Z) = 0$. It is straightforward to check that sig is a consistent signature assignment on $\mathcal{P}$. □

We now turn to another important consequence of the existence of a consistent signature assignment.

Proposition 3.20. *Given a locally-consistent annotated structure and a model $\mathcal{M}$ based on it, if a consistent signature assignment for such structure exists then, for each formula ψ annotating state s, $\mathcal{M}, s \models \psi$.*

Proof. Suppose sig is a consistent signature assignment for the given locally-consistent annotated structure $\langle \mathcal{S}, \Delta \rangle$. First, for each signature σ, define the valuation $\mathcal{V}_\sigma$ as follows:

$$\mathcal{V}_\sigma(X) \quad = \quad \{s \in S \mid X \in \Delta(s), \mathrm{sig}(s, X) \restriction X \leq \sigma \restriction X\}.$$

One simple observation is that $\sigma \leq \sigma'$ implies that $\mathcal{V}_\sigma(X) \subseteq \mathcal{V}_{\sigma'}(X)$ for each variable X. Moreover, if $\sigma \restriction Y \leq \sigma' \restriction Y$, then $\mathcal{V}_\sigma(X) \subseteq \mathcal{V}_{\sigma'}(X)$ for each variable X higher than or equal to Y.

Lemma 3.21. *If ψ annotates s then $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi$.*

Proof. Use induction on ψ. We shall refer to the condition SA1 - SA8 in Definition 3.13 in the proof.

- $\psi = P, \neg P$. Follows from the fact that $\mathcal{M}$ is based on $\langle \mathcal{S}, \Delta \rangle$.
- $\psi = X$. Follows from the definition of $\mathcal{V}_{\mathrm{sig}(s,X)}(X)$.
- $\psi = \psi_1 \vee \psi_2$. By condition SA1, for some i, ψ_i annotates s and $\mathrm{sig}(s, \psi_i) \leq \mathrm{sig}(s, \psi)$. By induction, $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi_i)}} \psi_i$ which implies that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi_i$. Thus $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi_1 \vee \psi_2$.
- $\psi = \psi_1 \wedge \psi_2$. By condition SA2, for each i, ψ_i annotates s and $\mathrm{sig}(s, \psi_i) \leq \mathrm{sig}(s, \psi)$. By induction, $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi_i)}} \psi_i$ which implies that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi_i$ for each i. Thus $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi_1 \wedge \psi_2$.
- $\psi = \langle a \rangle \psi'$. By condition SA3, for some $t \in R_a(s)$, ψ' annotates t and $\mathrm{sig}(t, \psi') \leq \mathrm{sig}(s, \psi)$. By induction, $\mathcal{M}, t \models_{\mathcal{V}_{\mathrm{sig}(t,\psi')}} \psi'$ which implies that $\mathcal{M}, t \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi'$. Thus $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \langle a \rangle \psi'$.
- $\psi = [a]\psi'$. By condition SA4, for each $t \in R_a(s)$, ψ' annotates t and $\mathrm{sig}(t, \psi') \leq \mathrm{sig}(s, \psi)$. By induction, $\mathcal{M}, t \models_{\mathcal{V}_{\mathrm{sig}(t,\psi')}} \psi'$ which implies that $\mathcal{M}, t \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi'$. Thus $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} [a]\psi'$.
- $\psi = \mu Z.\psi'$. We first prove the following claim:

  ($\star$) for any signature σ and state s, if Z annotates s and $\mathrm{sig}(s, Z) = \sigma$ then $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \mu Z.\psi'$.

  Assume otherwise and let σ be the *least* signature such that, for some state s annotated by Z, $\mathrm{sig}(s, Z) = \sigma$ but $\mathcal{M}, s \not\models_{\mathcal{V}_\sigma} \mu Z.\psi'$. Since Z annotates s, so does ψ'. Suppose $\mathrm{sig}(s, \psi') = \sigma'$; hence, by condition SA6, $\sigma' \restriction Z < \sigma \restriction Z$. By induction, we know that $\mathcal{M}, s \models_{\mathcal{V}_{\sigma'}} \psi'$. It can be shown that $\mathcal{M}, s \models_{\mathcal{V}_{\sigma'}} \mu Z.\psi'$. If this is *not* the case, since $\mathcal{M}, s \models_{\mathcal{V}_{\sigma'}} \psi'$, there must be some state $t \in \mathcal{V}_{\sigma'}(Z)$ such that $\mathcal{M}, t \not\models_{\mathcal{V}_{\sigma'}} \mu Z.\psi'$. By definition, $t \in \mathcal{V}_{\sigma'}(Z)$ means that Z annotates t and $\mathrm{sig}(t, Z) \restriction Z \leq \sigma' \restriction Z < \sigma \restriction Z$. It follows that $\mathcal{M}, t \not\models_{\mathcal{V}_{\mathrm{sig}(t,Z)}}$

$\mu Z.\psi'$. But this would mean that the signature $\mathrm{sig}(t, Z)$ fails the above claim, contradicting the assumption that σ is the least signature that does so. Hence we know that $\mathcal{M}, s \models_{\mathcal{V}_{\sigma'}} \mu Z.\psi'$. Since $\sigma' \restriction Z < \sigma \restriction Z$, it follows that $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \mu Z.\psi'$ contradicting the assumption that the claim fails with respect to signature σ. Therefore we may conclude that $(\star)$ holds.

Back to the main proof, we need to show that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \mu Z.\psi'$. Since Z annotates s, by $(\star)$, we know that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,Z)}} \mu Z.\psi'$. By condition SA5, $\mathrm{sig}(s, \psi) \restriction X \geq \mathrm{sig}(s, Z) \restriction X$ for each variable X higher than Z. This implies that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \mu Z.\psi'$ as required.

- $\psi = \nu X.\psi'$. We first prove the following claim:

  $(\star\star)$ for any signature σ and state s, if X annotates s and $\mathrm{sig}(s, X) = \sigma$ then $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \nu X.\psi'$.

  Suppose σ is a signature. For each state s in $\mathcal{V}_\sigma(X)$, X annotates s and $\mathrm{sig}(s, X) \restriction X \leq \sigma \restriction X$. Clearly ψ' also annotates s and, by condition SA8, $\mathrm{sig}(s, \psi') \restriction X \leq \mathrm{sig}(s, X) \restriction X$. Hence by induction we know that $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \psi'$. This means that $\mathcal{V}_\sigma(X) \subseteq \|\psi'\|_{\mathcal{V}_\sigma}$ which implies that $\mathcal{V}_\sigma(X) \subseteq \|\nu X.\psi'\|_{\mathcal{V}_\sigma}$. In other words, for any state s in $\mathcal{V}_\sigma(X)$, $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \nu X.\psi'$. This clearly implies the above claim.

  Back to the main proof, we need to show that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \nu X.\psi'$. Since X annotates s, by $(\star\star)$, we know that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,X)}} \nu X.\psi'$. By condition SA7, $\mathrm{sig}(s, \psi) \restriction X \geq \mathrm{sig}(s, X) \restriction X$. This implies that $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \nu X.\psi'$ as required.

□

Back to the proof of the proposition, we need to show that, for each formula ψ annotating s, $\mathcal{M}, s \models_{\mathcal{V}} \psi$ (where $\mathcal{V}$ is the valuation for the fixpoint variables of ϕ on $\mathcal{M}$; see Definition 3.2). We first show that, for any signature σ, $\mathcal{V}$ extends $\mathcal{V}_\sigma$, i.e. $\mathcal{V}_\sigma(X) \subseteq \mathcal{V}(X)$ for all variable X. Consider the sequence $X_1, ..., X_m$ of all the variables. We use induction on i $(1 \leq i \leq m)$ to show that $\mathcal{V}_\sigma(X_i) \subseteq \mathcal{V}(X_i)$. Assume the hypothesis holds at $< i$. Consider the case where X_i is a μ-variable. Suppose s is a state in $\mathcal{V}_\sigma(X_i)$; hence $\mathrm{sig}(s, X_i) \restriction X_i \leq \sigma \restriction X_i$. By $(\star)$ in the proof of the previous lemma, it follows that $\mathcal{M}, s \models_{\mathcal{V}_\sigma} \mu X_i.\psi_i$. But by the induction hypothesis we know that $\mathcal{V}_\sigma(X_j) \subseteq \mathcal{V}(X_j)$ for each $j < i$. This implies that $\mathcal{M}, s \models_{\mathcal{V}} \mu X_i.\psi_i$. Therefore $\mathcal{V}_\sigma(X_i) \subseteq \mathcal{V}(X_i)$. The case where X_i is a ν-variable can be shown similarly but using $(\star\star)$ instead of $(\star)$.

By the previous lemma, if ψ annotates s then $\mathcal{M}, s \models_{\mathcal{V}_{\mathrm{sig}(s,\psi)}} \psi$. Since $\mathcal{V}$ extends $\mathcal{V}_{\mathrm{sig}(s,\psi)}$, it follows that $\mathcal{M}, s \models_{\mathcal{V}} \psi$ as required. □

3.3 Fundamental Semantic Theorem

From the results on signature assignments shown previously, we immediately obtain the Fundamental Semantic Theorem of the Modal μ-Calculus.

Theorem 3.22 ([SE89])**.** *For any well-founded pre-model $\mathcal{P}$ and any model $\mathcal{M}$ based on $\mathcal{P}$, every formula annotating a state in $\mathcal{P}$ is true at that state in $\mathcal{M}$.*

Proof. By Proposition 3.16, every well-founded pre-model has a consistent signature assignment. By Proposition 3.20, $\mathcal{M}, s \models \psi$ for each ψ annotating state s. □

To prove the converse of the theorem (i.e. every annotated model has a well-founded dependency relation), we construct a consistent signature assignment from the model. We first explain some terminology which will be used extensively later.

Let $\mathcal{M}$ be a model. The valuation $\mathcal{V}_{\mathcal{M},\phi}$ (see Definition 3.2) can be *relativised* by a signature.

Definition 3.23. Given a signature $\sigma = \langle \alpha_1, ..., \alpha_n \rangle$, the *relativised valuation* $\mathcal{V}^\sigma_{\mathcal{M},\phi}$ is defined to be $\mathcal{V}^\sigma_m$, where $\mathcal{V}^\sigma_0, ..., \mathcal{V}^\sigma_m$ are defined iteratively as follows:

- $\mathcal{V}^\sigma_0(X) = \emptyset$ for all variables X;
- $\mathcal{V}^\sigma_{i+1} = \mathcal{V}^\sigma_i[X_{i+1} := \|\mu^{\alpha_j} Z_j.\psi\|_{\mathcal{V}^\sigma_i}]$, if X_{i+1} identifies $\mu Z_j.\psi$; and
- $\mathcal{V}^\sigma_{i+1} = \mathcal{V}^\sigma_i[X_{i+1} := \|\nu X_{i+1}.\psi\|_{\mathcal{V}^\sigma_i}]$, if X_{i+1} identifies $\nu X_{i+1}.\psi$.

Obviously, the valuation $\mathcal{V}_{\mathcal{M},\phi}$ *extends*[1] the relativised valuation $\mathcal{V}^\sigma$ for any signature σ. For brevity, we omit the subscript $\mathcal{M}, \phi$ whenever possible, and also write $\mathcal{M}, s \models_\sigma \psi$ for $\mathcal{M}, s \models_{\mathcal{V}^\sigma_{\mathcal{M},\phi}} \psi$.

As shown in Proposition 2.12(a), if a μ-formula $\mu Z.\psi$ is true at state s in the model, there must be a *least* ordinal α such that the approximant $\mu^\alpha Z.\psi$ is true at s. It is then not surprising to know that if we are given a formula ψ which is true at s, there must be a sequence of ordinals $\langle \alpha_1, ..., \alpha_n \rangle$ such that ψ is also true at s when each variable Z_i is interpreted as the α_i-approximant $\mu^{\alpha_i} Z_i.\psi_i$. We can state this precisely using relativised valuations.

Lemma 3.24. *For each formula ψ, if $\mathcal{M}, s \models \psi$ then there exists a signature σ such that $\mathcal{M}, s \models_\sigma \psi$.*

Proof. We can show that there exists a signature σ such that $\mathcal{V}^\sigma_i(X_j) = \mathcal{V}_i(X_j)$ for each i and $j \leq i$ (where $\mathcal{V}^\sigma_i$ and $\mathcal{V}_i$ are as defined in Definition 3.23 and 3.2).

Use induction on i. The base case ($i = 0$) is obvious. Assume this is true up to i. By induction, there is a signature σ such that $\mathcal{V}^\sigma_i(X_j) = \mathcal{V}_i(X_j)$ for each $j \leq i$. By definition, $\mathcal{V}^\sigma_{i+1}(X_j) = \mathcal{V}_{i+1}(X_j)$ for each $j \leq i$. Consider $\mathcal{V}^\sigma_{i+1}(X_{i+1})$. If X_{i+1} is a ν-variable, by definition, this equals to $\mathcal{V}_{i+1}(X_{i+1})$ as required.

On the other hand, if X_{i+1} identifies a formula $\mu X_{i+1}.\psi$, there must be a (least) ordinal β such that

$$\|\mu X_{i+1}.\psi\|_{\mathcal{V}^\sigma_i} \quad = \quad \|\mu^\beta X_{i+1}.\psi\|_{\mathcal{V}^\sigma_i}.$$

In this case, let $\sigma' = \sigma[X_{i+1} := \beta]$. Clearly, $\mathcal{V}^{\sigma'}_{i+1}(X_j) = \mathcal{V}_{i+1}(X_j)$ for each $j \leq i+1$, as required. □

[1] A valuation $\mathcal{V}$ is said to extend a valuation $\mathcal{V}'$ iff $\mathcal{V}'(X) \subseteq \mathcal{V}(X)$ for each variable X.

Definition 3.25. We define the *canonical signature assignment* on $\mathcal{M}$ to be the signature assignment $\mathrm{Sig}_{\mathcal{M},\phi}$ (on the annotated model $\langle \mathcal{M}, \Delta_{\mathcal{M},\phi} \rangle$) such that, for each state s and formula ψ true at s,

$$\mathrm{Sig}_{\mathcal{M},\phi}(s,\psi) \quad = \quad \bigwedge \{\sigma \mid \mathcal{M}, s \models_\sigma \psi\},$$

where $\bigwedge$ denotes the *lexicographically* least signature in the set.

$\mathrm{Sig}_{\mathcal{M},\phi}(s,\psi)$ is called the *signature of ψ at state s in $\mathcal{M}$.*

Observe that if a μ-variable Z is *not* active in ψ, then $\mathrm{Sig}_{\mathcal{M},\phi}(s,\psi)(Z) = 0$. It is straightforward to check that $\mathrm{Sig}_{\mathcal{M},\phi}$ is a consistent signature assignment. As usual, the subscript $\mathcal{M}, \phi$ is omitted whenever possible.

Lemma 3.26. *$\mathrm{Sig}_{\mathcal{M},\phi}$ is a consistent signature assignment on the annotated model $\langle \mathcal{M}, \Delta_{\mathcal{M},\phi} \rangle$.*

Proof. Straightforward. □

By Proposition 3.14, we obtain the converse of the Fundamental Semantic Theorem.

Theorem 3.27. *For any model $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}} \rangle$ and formula ϕ, there exists a dependency relation $\rightarrow$ such that $\langle \mathcal{S}, \Delta_{\mathcal{M},\phi}, \rightarrow \rangle$ is a well-founded pre-model.*

Proof. By Lemma 3.26, $\mathrm{Sig}_{\mathcal{M},\phi}$ is a consistent signature assignment on the annotated model $\langle \mathcal{M}, \Delta_{\mathcal{M},\phi} \rangle$. By Proposition 3.14, there exists a dependency relation $\rightarrow$ such that $\langle \mathcal{S}, \Delta_{\mathcal{M},\phi}, \rightarrow \rangle$ is a well-founded pre-model. □

The Fundamental Semantic Theorem and its converse greatly help us understand the logic. With these results, it becomes much easier to determine whether a formula is true in a model or not. As we study in the next chapter, this characterisation also enables us to manipulate models beyond what the basic techniques in modal logic can do.

3.3.1 Some Applications

Tree model property. In the previous chapter, we show by means of bisimulation equivalence that every model of a formula ϕ can be unravelled into a tree model for ϕ. From the Fundamental Semantic Theorem, we can "prune" such tree model to obtain a tree model whose degree is bounded by the number of $\langle\cdot\rangle$-subformulae of ϕ.

Theorem 3.28 (Tree Model Property)**.** *Every satisfiable formula ϕ has a tree model whose degree is bounded by the number of $\langle\cdot\rangle$-formulae in* $\mathrm{Sub}(\phi)$.

Proof. Without loss of generality, we assume that ϕ is a closed formula in positive normal form. By Proposition 2.27, ϕ has a tree model $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}} \rangle$. Let Δ be the canonical annotation of the subformulae of ϕ on $\mathcal{M}$. By Theorem 3.27, there exists a dependency relation $\rightarrow$ such that $\langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a well-founded pre-model.

For each state s in $\mathcal{S}$ and formula $\langle a\rangle\psi \in \Delta(s)$, there must be states $s' \in R_a(s)$ such that $\psi \in \Delta(s')$ and $(s, \langle a\rangle\psi) \to (s', \psi)$; we select $s_{\langle a\rangle\psi}$ from one of these state s'. Then we mark each child state t of s which is not selected (i.e. $t \neq s_{\langle a\rangle\psi}$ for any $\langle a\rangle\psi \in \Delta(s)$) for deletion. We obtain a new transition system $\mathcal{S}'$ by removing all the marked states and all their descendants (obviously, the root state is not removed). $\mathcal{S}'$ is a tree structure whose degree is bounded by the number of $\langle\cdot\rangle$-subformulae of ϕ. It is easy to check that the structure $\langle \mathcal{S}', \Delta', \to'\rangle$ (where Δ' and $\to'$ are the restrictions of Δ and $\to$ to the remaining states respectively) is still a well-founded pre-model. By Theorem 3.22, any model based on this new pre-model is a tree model for ϕ. □

$[\cdot]$-free formulae. Next we consider some applications of consistent signature assignments. First, we show that every $[\cdot]$-free formula, i.e. a formula not containing any $[\cdot]$-subformula, if satisfiable, has a model whose size is bounded by the number of $\langle\cdot\rangle$-subformulae. This generalises a similar result in modal logic.

Theorem 3.29. *Every satisfiable $[\cdot]$-free formula has a model in which the number of states is linear in the number of $\langle\cdot\rangle$-subformulae (hence, linear in the length of the formula).*

Proof. Suppose ϕ is a satisfiable $[\cdot]$-free formula. Let $\mathcal{M}$ be a model of ϕ. We assume the canonical annotation Δ of the subformulae of ϕ on $\mathcal{M}$. As mentioned earlier, there is a consistent signature assignment sig on $\mathcal{M}$ (of course, one possible signature assignment is the canonical one Sig. But any signature assignment satisfying the consistency conditions will do.)

Since ϕ is true in $\mathcal{M}$, there must be some state s_0 annotated with ϕ. Moreover, let $\langle a_1\rangle\psi_1, ..., \langle a_n\rangle\psi_n$ $(n \geq 0)$ be all the $\langle\cdot\rangle$-subformulae of ϕ which are true at some states. For each i, choose a state s_i annotated with ψ_i (there must be one because $\langle a_i\rangle\psi_i$ is true at some state) which has the least signature, i.e. $\mathrm{sig}(s_i, \psi_i) \leq \mathrm{sig}(s, \psi_i)$ for all state s annotated with ψ_i.

Define a transition system $\mathcal{S}' = \langle S', \{R'_a\}_{a\in \mathrm{Act}}\rangle$ where

- $S' = \{s_0, s_1, ..., s_n\}$,
- $s_j R_a s_i$ iff s_j is annotated with $\langle a\rangle\psi_i$.

We can show that the model $\mathcal{M}' = \langle \mathcal{S}', \mathcal{V}'_{\mathrm{Prop}}\rangle$ satisfies ϕ, where $\mathcal{V}'_{\mathrm{Prop}}$ is the restriction of $\mathcal{V}_{\mathrm{Prop}}$ to $\mathcal{S}'$.

Let Δ' be the restriction of Δ to the states in the new model. Since ϕ does *not* contain $[\cdot]$-subformulae, it is clear that Δ' is a locally-consistent annotation on $\mathcal{S}'$. More importantly, the signature assignment sig is locally consistent in the annotated structure $\langle \mathcal{S}', \Delta'\rangle$. The only condition we need to check is

- For each state s in $\mathcal{S}'$ annotated with $\langle a_i\rangle\psi_i$, there exists a state $t \in R'_{a_i}(s)$ annotated with ψ such that $\mathrm{sig}(t, \psi_i) \leq \mathrm{sig}(s, \langle a_i\rangle\psi_i)$.

From the above definition, we know that $s_i \in R'_{a_i}(s)$. Since sig is locally consistent in the original model, there must some state $s' \in R_{a_i}(s)$ annotated with ψ_i such that $\mathrm{sig}(s', \psi_i) \leq \mathrm{sig}(s, \langle a_i \rangle \psi_i)$. But by definition $\mathrm{sig}(s_i, \psi_i) \leq \mathrm{sig}(s', \psi_i)$. Hence $\mathrm{sig}(s_i, \psi_i) \leq \mathrm{sig}(s, \langle a_i \rangle \psi_i)$ as required.

Since $\langle \mathcal{S}', \Delta' \rangle$ has a consistent signature assignment, by Proposition 3.20, any model based on it, including $\mathcal{M}'$, satisfies the annotating formulae. This means that ϕ is true at s_0 in the new model. Clearly the number of states in the new model is no greater than $D + 1$, where D is the number of $\langle \cdot \rangle$-subformulae of ϕ. □

Finite model property. An important application of signatures is the *finite model property* of the modal μ-calculus. This follows from a well-known result in the theory of well quasi-ordering. This result was first shown in [Koz86]. Using signatures, we can present a much clearer proof than the original one.

Let $\mathcal{M} = \langle S, \{R_a\}_{a \in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}} \rangle$ be a model and Δ be the canonical annotation of the subformulae of ϕ on $\mathcal{M}$. Suppose sig is a signature assignment on $\langle \mathcal{M}, \Delta \rangle$. Define an ordering $\sqsubseteq$ over states: $s \sqsubseteq t$ iff

- $\Delta(s) = \Delta(t)$,
- $\mathrm{sig}(s, \psi) \leq \mathrm{sig}(t, \psi)$ for each formula ψ annotating s.

Since the lexicographical ordering $\leq$ on signatures is a well ordering and there are finitely many subformulae of ϕ, it follows that $\sqsubseteq$ is a well quasi-ordering (see Definition 0.3).

Lemma 3.30. *$\langle S, \sqsubseteq \rangle$ is a well quasi-ordered set.*

Proof. For each subformula ψ of ϕ, define an ordering $\sqsubseteq_\psi$ on S: $s \sqsubseteq_\psi t$ iff either both $\Delta(s)$ and $\Delta(t)$ do not contain ψ, or they both contain ψ and $\mathrm{sig}(s, \psi) \leq \mathrm{sig}(t, \psi)$. Clearly $s \sqsubseteq t$ iff $s \sqsubseteq_\psi t$ for each subformula ψ of ϕ. It is easily seen that $\langle S, \sqsubseteq \rangle$ is (isomorphic to) a subset of the product of the finite family $\langle S, \sqsubseteq_\psi \rangle$, $\psi \in \mathrm{Sub}(\phi)$. Each $\langle S, \sqsubseteq_\psi \rangle$ is a well quasi-ordered set (because $\leq$ is a well ordering on signatures). By Proposition 0.4, $\langle S, \sqsubseteq \rangle$ is also a well quasi-ordered set. □

From a model $\mathcal{M}$ of ϕ and a well quasi-ordering $\sqsubseteq$ given by a *consistent* signature assignment, a finite model for ϕ can be obtained by taking the quotient of $\mathcal{M}$ with respect to $\sqsubseteq$. The quotient construction can be avoided by defining a finite model from the states in a finite *base* of $\langle S, \sqsubseteq \rangle$ directly.

Theorem 3.31 (Finite Model Theorem [Koz86])**.** *Every satisfiable modal μ-calculus formula has a finite model.*

Proof. Let ϕ be a satisfiable formula and $\mathcal{M} = \langle S, \{R_a\}_{a \in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}} \rangle$ be a model for ϕ. We assume the canonical annotation Δ of the subformulae of ϕ on $\mathcal{M}$. As mentioned earlier, there is a consistent signature assignment sig on $\mathcal{M}$. The ordering $\sqsubseteq$ is defined

with respect to sig as above. Since $\langle S, \sqsubseteq \rangle$ is a well quasi-ordered set, S must have a *finite base* $S' \subseteq S$: for all $s \in S$ there exists $t \in S'$ such that $t \sqsubseteq s$ (see Definition 0.3).

Define a model $\mathcal{M}' = \langle S', \{R'_a\}_{a \in \mathrm{Act}}, \mathcal{V}'_{\mathrm{Prop}} \rangle$ where

- S' is some finite base of S,
- sR'_at iff, for some $s' \in S$, sR_as' and $t \sqsubseteq s'$,
- $\mathcal{V}'_{\mathrm{Prop}}$ is the restriction of $\mathcal{V}_{\mathrm{Prop}}$ to states in S'.

$\mathcal{M}'$ is finite by definition. Suppose ϕ is true at state s_0 in $\mathcal{M}$. There must be some state $s'_0 \sqsubseteq s_0$ in S'. We can show that ϕ is true at s'_0 in $\mathcal{M}'$.

Let Δ' be the restriction of Δ to the states in the new model. It is clear from the definition that Δ' is locally consistent on $\mathcal{M}'$. More importantly, we can show that sig is a locally-consistent signature assignment on the new annotated structure $\langle \mathcal{M}', \Delta' \rangle$. The two conditions we need to check are:

- For each state $s \in S'$ annotated by $\langle a \rangle \psi$, there exists a state $t \in R'_a(s)$ annotated with ψ such that $\mathrm{sig}(t, \psi) \leq \mathrm{sig}(s, \langle a \rangle \psi)$.
- For each state $s \in S'$ annotated by $[a]\psi$ and each state $t \in R'_a(s)$, ψ annotates t and $\mathrm{sig}(t, \psi) \leq \mathrm{sig}(s, [a]\psi)$.

Consider the first case. Let s be a state in S' annotated by $\langle a \rangle \psi$. There must be a state $s' \in R_a(s)$ annotated by ψ such that $\mathrm{sig}(s', \psi) \leq \mathrm{sig}(s, \langle a \rangle \psi)$. Let t be a state in S' such that $t \sqsubseteq s'$; hence $\mathrm{sig}(t, \psi) \leq \mathrm{sig}(s', \psi)$. This implies that, $t \in R'_a(s)$ and $\mathrm{sig}(t, \psi) \leq \mathrm{sig}(s, \langle a \rangle \psi)$ as required. The second case can be shown similarly.

Since $\langle \mathcal{M}', \Delta' \rangle$ has a consistent signature assignment, by Proposition 3.20, $\mathcal{M}'$ satisfies the annotating formulae. Since ϕ annotates s_0 and $s'_0 \sqsubseteq s_0$, it follows that ϕ is true at s'_0 in $\mathcal{M}'$. □

3.4 Tableau Methods

Tableau methods have long been used for solving problems related to logics; the validity, satisfiability, and model checking problems, in particular. For traditional logics such as first-order logic and various modal logics, tableau techniques have been extensively studied ([Smu68],[HC68],[Fit83]) and have now become a standard tool for solving logical problems. More recently, tableau techniques have been used for the modal and temporal logics of computational systems. In these areas, tableau methods are typically used for model checking, i.e. determining whether a formula is true at a state in the system. An early work is a tableau system by Stirling [Sti87] which is used for a modal logic for concurrent systems represented in CCS ([Mil80],[Mil89]). Stirling's tableau techniques have then been extended for model checking in the logics with fixpoints ([Lar90],[Win89]) and the modal μ-calculus and related logics ([SW91],[Cle90]). With their use in model checking, tableau techniques have become important tools in system verification.

The use of tableaux offers many advantages. A tableau system can be implemented

(i.e. a program which tries to find a successful tableau proof is written) or used for proving properties of the logic. Generally, the existence of a tableau system in which a successful tableau is a (boundedly) finite structure implies that the logical problem that the tableau system is designed for (e.g. model checking, satisfiability, etc.) is decidable. The computational complexity of the problem can often be derived from the structure of a tableau. From the theoretical point of view, tableaux can be used as a tool to study and prove related properties of the logic. For example, a tableau system might be used to establish the completeness of an axiomatisation of the logic, as is done for various modal logics.

In contrast to the aforementioned works on tableaux for model checking, we are interested in the tableau system for the satisfiability problem of the modal μ-calculus. Compared to model checking, such tableau systems are not prevalent in literature. The reason might be the less practical interest in the satisfiability problem, the emergence of the automata-theoretic technique [SE89], or the difficulty of the satisfiability problem itself. In the original paper by Kozen [Koz83], a tableau system for checking the satisfiability of a fragment of the modal μ-calculus (called *aconjunctive formulae*) is given and used to establish a complexity result and the completeness of an axiomatisation for such fragment. Niwiński and Walukiewicz [NW97] propose a tableau system for the full modal μ-calculus mainly for proving the completeness of a deductive system. The idea behind Niwiński and Walukiewicz's tableaux is closely related to Streett and Emerson's notion of well-founded pre-models. From the computational aspect, their tableau system is of limited use because a tableau in such system is generally an infinite tree structure. Hence one cannot obtain the decidability nor the complexity result solely from it. However, Niwiński and Walukiewicz's work is a good starting point for designing a finitistic tableau system. We describe a tableau proof system based on their work below.

3.4.1 Tableau Systems

A *tableau system* is a goal-directed proof system. The main ingredients are the *tableau rules*, the *termination condition*, and the *success condition*. The exact structure of a *goal* is also specified in the tableau system. In our case where a tableau system is used for showing satisfiability, a goal typically contains a set of formulae for which the satisfiability are to be established, and possibly some extra information. Tableau rules are generally of the form:

$$\mathsf{R}: \frac{G}{G_1 \,|\, ... \,|\, G_n} \quad C$$

where $n \geq 1$[2], $G, G_1, ..., G_n$ are goal schemata and C is a side condition which restricts the application of the rule. These rules are generally backward sound in the sense that if all the goals $G_1, ..., G_n$ are "true", then so is the goal G. Basically, to obtain a *tableau*,

[2]It is possible to include a tableau rule in which $n = 0$. We do not use such tableau rule in this thesis.

one starts with an *initial goal* and subsequently apply one of the applicable tableau rules. The termination condition specifies when a node is considered a *terminal*, which is *not* expanded further. The *success condition* specifies when a fully-expanded tableau is considered *successful*.

The underlying structure of a tableau is not particularly important. However, to be precise, we define a tableau formally as follows. A *tableau* is a tree $\mathcal{T}$ satisfying the following:

- Each node u in $\mathcal{T}$ is labelled by a pair of the form (g, R) or $(g, \cdot)$, where g is a goal and R is the name of a tableau rule (g is referred to as the *goal at* u and R as the *tableau rule applied at* u).
- For each node u labelled by (g, R),
  - there is an instance $\frac{g}{g_1 \mid \ldots \mid g_n} c$ of rule R such that the condition c is satisfied,
  - u has n children $u_1, ..., u_n$ whose goals are $g_1, ..., g_n$ respectively ($g_1, ..., g_n$ are referred to as the *subgoals* of goal g at u),
  - u is not a *terminal* (as specified by the termination condition).
- Each node u labelled by $(g, \cdot)$ is a leaf in $\mathcal{T}$.

A *maximal* tableau (also called *fully-expanded* tableau) is a tableau in which each leaf is a terminal. A *successful tableau* is a maximal tableau satisfying the success condition.

In our tableau systems for satisfiability checking, a successful tableau can be seen as a model of the given formula. Hence constructing a successful tableau amounts to constructing a model.

3.4.2 Tableau System TS_0

We now describe a tableau system, called TS_0, which is based on the tableau method in [NW97]. In this tableau system, we assume that we are given a closed and *guarded* formula ϕ. The closedness assumption is introduced for convenience (so that there is no need to distinguish the free variables and the bound variables). The reason we require the formula be guarded is due to some complication which arises when a fixpoint formula can be unfolded indefinitely without reaching a modal formula (an obvious example being $\sigma X.X$). As we shall explain later, the tableau system can be modified to cope with unguarded formulae.

A *goal* in a TS_0-tableau for ϕ is a set of subformulae of ϕ. The initial goal is $\{\phi\}$. We typically omit the parentheses when writing goals (e.g. $\psi_1, ..., \psi_n$ and $\psi_1, ..., \psi_n, \Gamma$, where Γ is a set of formulae, stand for $\{\psi_1, ..., \psi_n\}$ and $\{\psi_1, ..., \psi_m\} \cup \Gamma$, respectively).

Tableau rules. The tableau rules are as follows:

$$\mathsf{R}\wedge: \quad \frac{\psi_1 \wedge \psi_2, \Gamma}{\psi_1, \psi_2, \Gamma}$$

$$\mathsf{R}\vee: \quad \frac{\psi_1 \vee \psi_2, \Gamma}{\psi_i, \Gamma} \quad i \in \{1, 2\}$$

$$\mathsf{R}\sigma: \quad \frac{\sigma X.\psi, \Gamma}{X, \Gamma} \quad \sigma \in \{\mu, \nu\}$$

$$\mathsf{Unfold}_\sigma: \quad \frac{X, \Gamma}{\psi, \Gamma} \quad X \text{ identifies } \sigma X.\psi$$

$$\mathsf{R}\langle\rangle: \quad \frac{\langle a_1\rangle\psi_1, ..., \langle a_n\rangle\psi_n, \Gamma}{\psi_1, \Gamma_{a_1} \mid ... \mid \psi_n, \Gamma_{a_n}} \quad n \geq 1$$

where

- Γ contains only literals and $[\cdot]$-formulae, and
- for each action a, $\Gamma_a = \{\psi \mid [a]\psi \in \Gamma\}$.

Termination. A *terminal* is a node satisfying *one* of the following:

T1. The goal contains a complementary pair of literals.

T2. No tableau rule is applicable. This is the case when the goal only contains literals and $[\cdot]$-formulae.

Clearly it is possible for a tableau to have an (infinite) branch which does not contain any terminal.

Success. To define when a tableau is considered successful, the notion of trails in tableaux is needed. This is similar to trails in pre-models. First, each tableau determines a dependency relation in the obvious way.

Definition 3.32. The *dependency relation* $\rightarrow$ *on* $\mathcal{T}$ is the smallest relation on pairs of the form (u, ψ), where u is a node and ψ is a formula in the goal at u, satisfying the following:

- For each node u where rule $\mathsf{R}\wedge$, $\mathsf{R}\vee$, $\mathsf{R}\sigma$, or Unfold_σ is applied, if a formula ψ at u is reduced to ψ' at its child u', then $(u, \psi) \rightarrow (u', \psi')$; but if ψ is *not* reduced by the rule (hence ψ is also in u'), then $(u, \psi) \rightarrow (u', \psi)$.
- For each node u where rule $\mathsf{R}\langle\rangle$ is applied and $\langle a_1\rangle\psi_1, ..., \langle a_n\rangle\psi_n, \Gamma$ are its formulae, if a child u_i is expanded from $\langle a_i\rangle\psi_i$, then $(u, \langle a_i\rangle\psi_i) \rightarrow (u_i, \psi_i)$ and, for each formula $[a_i]\psi \in \Gamma$, $(u, [a_i]\psi) \rightarrow (u_i, \psi)$.

A *trail* in tableau $\mathcal{T}$ is a path over its dependency relation. Terminology on trails in pre-models can be given here in the obvious way. In particular, a μ*-trail* (ν-trail) is

an infinite trail in which the outermost variable unfolded infinitely often is a μ-variable (respectively ν-variable).

A tableau is said to be *successful* iff

S1. each leaf satisfies T2 but *not* T1 (hence the goal at each leaf is a consistent set of literals and $[\cdot]$-formulae), and

S2. every infinite trail is a ν-trail.

In other words, a successful tableau is a *maximal* tableau in which each node does *not* contain complementary literals *and* no μ-trail exists.

A successful tableau for ϕ can be seen as a well-founded (tree) pre-model. Conversely, from a well-founded pre-model where ϕ annotates some state, we can easily construct a successful tableau for ϕ. From the result previously shown, it is not surprising that this tableau system is sound and complete. We briefly outline the soundness and the completeness proof below.

Soundness. Suppose $\mathcal{T}$ is a successful tableau for ϕ. A model for ϕ can be constructed by identifying each "modal node" as a state. A *modal node* is either a node where rule $\mathsf{R}\langle\rangle$ is applied or a leaf node. For clarity, we use the letters s, t and their scripted versions to denote modal nodes. The following lemma is the reason why ϕ is required to be guarded.

Lemma 3.33. *For each node u, there is a* unique *modal node s below (or equal to) u such that rule $\mathsf{R}\langle\rangle$ is not applied in between.*

Proof. Follows from the guardedness assumption of ϕ. □

For each modal node s, let $[s]$ be the set of all ancestors u of s such that rule $\mathsf{R}\langle\rangle$ is *not* applied between u and s. By the previous lemma and the fact that every tableau rule other than $\mathsf{R}\langle\rangle$ creates *one* child, every node u belongs to a unique set $[s]$.

Definition 3.34. Define $\mathcal{M}_{\mathcal{T}} = \langle S, \{R_a\}_{a\in\mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}}\rangle$ to be the model where

- S is the set of all modal nodes in $\mathcal{T}$,
- sR_at iff there is a child u of s such that a formula $\langle a\rangle\psi$ at s is reduced to ψ at u, and $u \in [t]$,
- $\mathcal{V}_{\mathrm{Prop}}(P) = \{s \mid P \text{ is in the goal at } s\}$.

Obviously $\mathcal{M}_{\mathcal{T}}$ is a tree model. It can be shown that $\mathcal{M}_{\mathcal{T}}$ satisfies ϕ. There are many ways to show this. We briefly explain some possibilities below.

Theorem 3.35 (Soundness of TS_0)**.** *Every* guarded *formula which has a successful TS_0-tableau is satisfiable.*

Proof. Suppose $\mathcal{T}$ is a successful tableau for the given guarded formula ϕ and $\mathcal{M}_{\mathcal{T}}$ is as defined above. It can be shown that, for each node u and each formula ψ at u, if u is in $[s]$ then $\mathcal{M}_{\mathcal{T}}, s \models \psi$. This implies that ϕ is true at the root of $\mathcal{M}_{\mathcal{T}}$.

One way is to define a locally-consistent signature assignment on $\mathcal{M}_\mathcal{T}$. First, for each pair (u, ψ) where u is a node and formula ψ is in u, we assign a signature $\mathrm{sig}_\mathcal{T}(u, \psi)$ as in Definition 3.17 and 3.19 (this is possible because there is no μ-trail in the tableau). Next we annotate each state s in $\mathcal{M}_\mathcal{T}$ with the set $\Delta(s)$ of all the formulae annotating some node in $[s]$. It is easy to check that Δ is a locally-consistent annotation (e.g. for each s, $\Delta(s)$ does not contain complementary literals, if $\psi_1 \vee \psi_2$ is in $\Delta(s)$ then so is some ψ_i, etc.) Note that this relies on the fact that, in a tableau for a guarded formula, every formula in a goal must eventually be reduced. We then define a signature assignment sig on the annotated structure $\langle \mathcal{M}_\mathcal{T}, \Delta \rangle$ such that $\mathrm{sig}(s, \psi)$ is the least signature in $\{\mathrm{sig}_\mathcal{T}(u, \psi) \mid u \in [s]\}$. It is then straightforward to show that sig is locally consistent. The above claim then follows from Proposition 3.20.

It is also possible to prove the claim without relying on previous results. To do so, we first define a valuation $\mathcal{V}_{u,\psi}$ for each node u and formula ψ in u:

> $\mathcal{V}_{u,\psi}(X) = \{s \mid$ for some node $v \in [s]$, there is a trail from (u, ψ) to (v, X) along which X is active $\}$.

It can then be shown that, for each node u and each formula ψ at u, if u is in $[s]$ then $\mathcal{M}_\mathcal{T}, s \models_{\mathcal{V}_{u,\psi}} \psi$. The proof is a straightforward induction on ψ and relies on the assumption that the tableau is successful. Since ϕ is closed, this immediately implies that ϕ is true at the root of $\mathcal{M}_\mathcal{T}$. □

Completeness. From a model for ϕ, it is easy to construct a successful tableau. This is similar to the construction of a well-founded pre-model from a model in Theorem 3.27. The key is to make the choices which minimise signatures. We explain this in more detail below. Note that the guardedness assumption of ϕ is *not* needed in the completeness proof.

Theorem 3.36 (Completeness of TS_0)**.** *Every satisfiable closed formula (in positive normal form) has a successful TS_0-tableau.*

Proof. Suppose ϕ is true at state s_0 in a model $\mathcal{M}$. We construct a tableau for ϕ where each goal is augmented with a state; an extended goal is written as $s \vdash \Gamma$. In the construction, we maintain the property that the formulae in a goal is true at the augmented state, i.e. if $s \vdash \Gamma$ is a goal then $\mathcal{M}, s \models \Gamma$.

The construction starts with $s_0 \vdash \phi$. We continue expanding the tableau by applying an applicable rule to each leaf. Rules $\mathsf{R}\wedge$, $\mathsf{R}\sigma$, and Unfold_σ can be applied (to some formula) in one way, and clearly the above property is preserved. Suppose we reach a goal $s \vdash \psi_1 \vee \psi_2, \Gamma$ and would like to apply rule $\mathsf{R}\vee$. Since $\mathcal{M}, s \models \psi_1 \vee \psi_2, \Gamma$, there must be some i such that $\mathcal{M}, s \models \psi_i, \Gamma$ and $\mathrm{Sig}(s, \psi_i) \leq \mathrm{Sig}(s, \psi_1 \vee \psi_2)$. We apply rule $\mathsf{R}\vee$ and choose such i. Similarly, suppose we reach a goal

$$s \vdash \langle a_1 \rangle \psi_1, ..., \langle a_n \rangle \psi_n, \Gamma$$

and rule $\mathsf{R}\langle\rangle$ is applicable. Since the above formulae are true at s, for each i, there must be some state $s_i \in R_{a_i}(s)$ such that $\mathcal{M}, s_i \models \psi_i, \Gamma_{a_i}$. We apply rule $\mathsf{R}\langle\rangle$ and create a subgoal $s_i \vdash \psi_i, \Gamma_{a_i}$ for each i.

Suppose $\mathcal{T}$ is a maximal tableau constructed as above. It is clear that if there is a trail from (u, ψ) to (u', ψ') along which a μ-variable Z is active, then $\mathrm{Sig}(s, \psi) \geq \mathrm{Sig}(s', \psi')$ where s and s' are the states associated with the goals at u and u' respectively; and if Z is unfolded along such trail then $\mathrm{Sig}(s, \psi) > \mathrm{Sig}(s', \psi')$. This can be shown as in Lemma 3.15. It follows that $\mathcal{T}$ cannot contain a μ-trail. Since the formulae in each goal is true at some state, no goal in $\mathcal{T}$ contains complementary literals. Hence $\mathcal{T}$ is a successful tableau for ϕ. □

3.4.3 Guardedness and Tableaux.

Unguarded formulae introduce some complication when developing a tableau system. A tableau for such formulae may contain a path where some variable is unfolded indefinitely while some other formulae in the goal are not reduced. For example, consider a tableau for the formula $(\nu X.X) \wedge (\mu Z.Z)$ which is clearly unsatisfiable. By unfolding only the variable X, we obtain a successful tableau below.

$$
\begin{array}{c}
(\nu X.X) \wedge (\mu Z.Z) \\
\nu X.X, \mu Z.Z \\
X, \mu Z.Z \\
X, \mu Z.Z \\
\vdots
\end{array}
$$

This shows that TS_0 is not sound for unguarded formulae. Clearly the problem only arises when there is an unguarded ν-formula.

To show the satisfiability of unguarded formulae, one possibility is to *guard* the formula first. Proposition 2.18 shows that every formula can be translated into a semantically-equivalent guarded one. It is also not difficult to modify a tableau system to cope with unguarded formulae directly. Basically, what we do is to allow a ν-variable to be unfolded only *once* between consecutive modal nodes (unless some higher variable is unfolded). One solution is to record on each formula its history of "recent" unfoldings of ν-variables. This is the method used in the tableau system in [Koz83]. We explain a modified tableau system, called TS'_0, in more detail below.

Tableau system TS'_0. A *goal* in a tableau is now a set of augmented formulae of the form ψ^S, where S is a set of ν-variables. It will be clear from the tableau rules that the variables in S are linear ordered (under $\preccurlyeq$).

The *initial goal* in a tableau for ϕ is $\phi^\emptyset$. The modified tableau rules are given below.

We use $S \restriction X$ to denote $\{Y \in S \mid Y \preccurlyeq X\}$.

$$\mathsf{R}\wedge: \quad \frac{(\psi_1 \wedge \psi_2)^S, \Gamma}{\psi_1^S, \psi_2^S, \Gamma}$$

$$\mathsf{R}\vee: \quad \frac{(\psi_1 \vee \psi_2)^S, \Gamma}{\psi_i^S, \Gamma} \quad i \in \{1,2\}$$

$$\mathsf{R}\sigma: \quad \frac{(\sigma X.\psi)^S, \Gamma}{X^S, \Gamma}$$

$$\mathsf{Unfold}_\mu: \quad \frac{Z^S, \Gamma}{\psi^{S \restriction Z}, \Gamma} \quad Z \text{ identifies } \mu Z.\psi.$$

$$\mathsf{Unfold}_\nu: \quad \frac{X^S, \Gamma}{\psi^{(S \restriction X) \cup \{X\}}, \Gamma} \quad X \text{ identifies } \nu X.\psi \text{ and } X \notin S.$$

$$\mathsf{R}\langle\rangle: \quad \frac{(\langle a_1 \rangle \psi_1)^{S_1}, ..., (\langle a_n \rangle \psi_n)^{S_n}, \Gamma}{\psi_1^\emptyset, \Gamma_{a_1}^\emptyset \mid ... \mid \psi_n^\emptyset, \Gamma_{a_n}^\emptyset} \quad n \geq 1$$

where

- Γ contains only literals, $[\cdot]$-formulae, and ν-variables X^S where $X \in S$, and
- for each action a, $\Gamma_a^\emptyset = \{\psi^\emptyset \mid ([a]\psi)^S \in \Gamma\}$.

The termination and success conditions are the same as in TS_0. The following are some simple properties of TS_0'-tableaux.

Lemma 3.37. *For any node u in a TS_0'-tableau $\mathcal{T}$, if X^S, where X is a ν-variable and $X \in S$, is in the goal at u, then there must be a node v above u such that*

- *$X^{S'}$ is unfolded at v for some S'* not *containing X; and*
- *there is a trail from $(v, X^{S'})$ to (u, X^S), and X is active in every such trail.*

Proof. Clearly, since $X \in S$, there must be a node v above u where $X^{S'}$ (for some S') is unfolded and there is a trail from $(v, X^{S'})$ to (u, X^S). Let v be the *lowest* such node. It is clear that X must be active in every trail from $(v, X^{S'})$ to (u, X^S) (for otherwise, there would be a variable Y higher than X unfolded on one of those trails, and consequently, there would be some node v' below v satisfying the condition). □

Recall that a *modal node* is either a terminal node or a node where rule $\mathsf{R}\langle\rangle$ is applied.

Lemma 3.38. *Every node in a successful TS_0'-tableau is an ancestor of some modal node.*

Proof. Suppose $\mathcal{T}$ is a successful TS_0'-tableau. Suppose there is a node u which is not an ancestor of any modal node. Hence, there must be an infinite trail from u which does not go through a modal node. Since $\mathcal{T}$ is successful, such trail must be a ν-trail, i.e. the outermost variable X unfolded infinitely often on this trail is a ν-variable. This means that there is a subtrail $(v, X^S) \to ... \to (v', X^{S'})$ where $X^{S'}$ is unfolded at v' and no higher variable occurs on the trail. Clearly, this implies that $X \in S'$, which violates the condition for applying rule Unfold_ν. □

This means that we can define a model $\mathcal{M}_\mathcal{T}$ from a successful tableau $\mathcal{T}$ for ϕ as in the soundness proof of TS_0.

Theorem 3.39 (Soundness of TS_0')**.** *Every formula which has a successful TS_0'-tableau is satisfiable.*

Proof. We show that $\mathcal{M}_\mathcal{T}$ is a model of ϕ in the same way as in TS_0. By Lemma 3.37, for each *modal node* u which contains a formula X^S (where $X \in S$), there is a node v above u and a formula $X^{S'}$ satisfying the condition in the lemma; call $(v, X^{S'})$ a *companion* of (u, X^S). We extend the dependency relation $\to$ by including an arrow from each (u, X^S) (where u is a modal node) to one of its companions $(v, X^{S'})$. Let $\to'$ denote this extended dependency relation. It is easily seen that $\to'$ is still well-founded, i.e. there is no μ-trail over $\to'$. The rest of the proof is as in Theorem 3.35, using the extended dependency relation $\to'$ instead of $\to$. □

Theorem 3.40 (Completeness of TS_0')**.** *Every satisfiable closed formula (in positive normal form) has a successful TS_0'-tableau.*

Proof. The proof is the same as for TS_0. □

To avoid unnecessary complication, we shall keep the guardedness assumption in all tableau systems in the thesis. The above solutions can be applied to those tableau systems to cope with unguarded formulae directly.

Chapter 4

Tableau Systems for the Modal μ-Calculus

The tableau system TS_0 explained in the last chapter is not an effective tool for solving the satisfiability problem. It is not clear how to determine whether a formula has a successful tableau in TS_0. Obviously, since a successful tableau may be potentially infinite, we cannot simply try to enumerate all possible tableaux. The difficulty lies in its success condition, which involves checking for an (infinite) μ-trail. As we mentioned earlier, TS_0 is based on the tableau system in [NW97], which was introduced mainly for solving the completeness of an axiomatisation [Wal00].

It is one of our main goals of research to find a better tableau system for checking satisfiability. The aim is to obtain a tableau system in which a tableau is a finite structure and the success of a tableau can be determined in a simple manner. The tableau system TS described in this chapter has such property. The underlying idea is to record a partial history of the derivation of each formula in the tableau, and then use the recorded information to determine whether the tableau is considered successful.

4.1 Motivation

Because of fixpoint formulae, a tableau in TS_0 may never terminate. But since there are finitely many distinct formulae in the tableau, an infinite branch must contain a number of repeating goals. In particular, an infinite tableau must contain some node u and node v above it with the same goal Γ:

$$\begin{array}{c} v : \Gamma \\ \vdots \\ u : \Gamma \end{array}$$

In certain cases, we may successfully terminate the branch at node u because, in the model based on this tableau (as given by Definition 3.34), we may "merge" the two

states corresponding to nodes u and v to obtain a model for the initial formula. Such an extra success condition can be stated in terms of trails. Basically, if adding a backedge from node u to its companion v does not introduce a μ-trail, then u may be declared as a successful terminal. Consider the following example.

Example 4.1. Let ϕ be the conjunction of the following formulae (which state that "P infinitely often" and "$\neg P$ infinitely often"):

$$\nu X_1.(\mu Z.P \vee \langle a \rangle Z) \wedge [a]X_1$$
$$\nu X_2.(\mu Y.\neg P \vee \langle a \rangle Y) \wedge [a]X_2$$

Consider a fragment of a tableau for ϕ shown in Figure 4.1(a). Notice that node 14 and node 7 have the same goal, and similarly for node 20 and node 6. We may declare node 20 as a successful terminal because, by adding a backedge from node 20 to node 6, no μ-trail is introduced and hence the corresponding model, as shown in Figure 4.1(b), is a model for ϕ.

On the other hand, although node 14 has the same goal as node 7, adding an edge between these two nodes introduces a μ-trail:

$$(7, Y) \rightarrow (8, \neg P \vee \langle a \rangle Y) \rightarrow (9, \langle a \rangle Y) \rightarrow (10, Y) \rightarrow (11, Y) \rightarrow ... \rightarrow (14, Y) \rightarrow (7, Y) \rightarrow ...$$

In fact, if we do add such an edge, the model corresponding to the tableau obtained will consist of one state which, clearly, does not satisfy ϕ. Hence, we need to delay the termination until node 20.

In addition, we may want to identify which leaf node with a repeating goal is considered *unsuccessful*. Suppose u is a leaf node in some tableau for ϕ and v is some node above u with the same goal. If it is known that the region between v and u is unnecessary for obtaining a successful tableau, then we may want to declare u as unsuccessful. This will help bounding the size of tableaux for ϕ. Consider the following simple example.

Example 4.2. Suppose ϕ is the following "P infinitely often" formula

$$\nu X.(\mu Z.P \vee [a]Z) \wedge \langle a \rangle X$$

Consider a fragment of a tableau for ϕ shown in Figure 4.2(a). Notice that node 5 has the same goal as node 11. However, it is clear that there is a successful tableau for ϕ which does not contain the region between these two nodes. One such tableau is shown in Figure 4.2(b). In the latter tableau, rule $\mathsf{R}\vee$ at node 6 makes a better choice, i.e. choosing P instead of $[a]Z$.

Thus our task is to identify the extra termination and success conditions so that a successful tableau is a finite structure which is free of μ-trails, and hence can be seen as a model of the initial formula. As mentioned, our approach involves recording a partial

1: $\underline{(\nu X_1.(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1) \wedge (\nu X_2.(\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2)}\ \mathsf{R}\wedge$

2: $\underline{\nu X_1.(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ \nu X_2.(\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{R}\nu$

3: $\underline{X_1,\ X_2}\ \mathsf{Unfold}_\nu$

4: $\underline{(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{R}\wedge$

5: $\underline{\mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu$

6: $\underline{Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu$

7: $\underline{Z,\ [a]X_1,\ Y,\ [a]X_2}\ \mathsf{Unfold}_\mu$

8: $\underline{P \vee \langle a\rangle Z,\ [a]X_1,\ \neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\vee$

9: $\underline{\underline{P,\ [a]X_1,\ \langle a\rangle Y,\ [a]X_2}}\ \mathsf{R}\langle\rangle$

10: $\underline{X_1,\ Y,\ X_2}\ \mathsf{Unfold}_\nu$

11: $\underline{(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ Y,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{R}\wedge$

12: $\underline{\mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ Y,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu$

13: $\underline{Z,\ [a]X_1,\ Y,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu$

14: $\underline{Z,\ [a]X_1,\ Y,\ [a]X_2}\ \mathsf{Unfold}_\mu$

15: $\underline{P \vee \langle a\rangle Z,\ [a]X_1,\ \neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\vee$

16: $\underline{\underline{\langle a\rangle Z,\ [a]X_1,\ \neg P,\ [a]X_2}}\ \mathsf{R}\langle\rangle$

17: $\underline{Z,\ X_1,\ X_2}\ \mathsf{Unfold}_\nu$

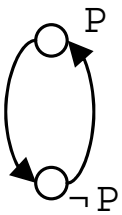
18: $\underline{Z,\ (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{R}\wedge$

19: $\underline{Z,\ \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu$

20: $\underline{Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}$

$\vdots$

(a)



(b)

Figure 4.1: A fragment of a tableau for ϕ in Example 4.1 and the corresponding model for the tableau when a backedge from node 20 to node 6 is added.

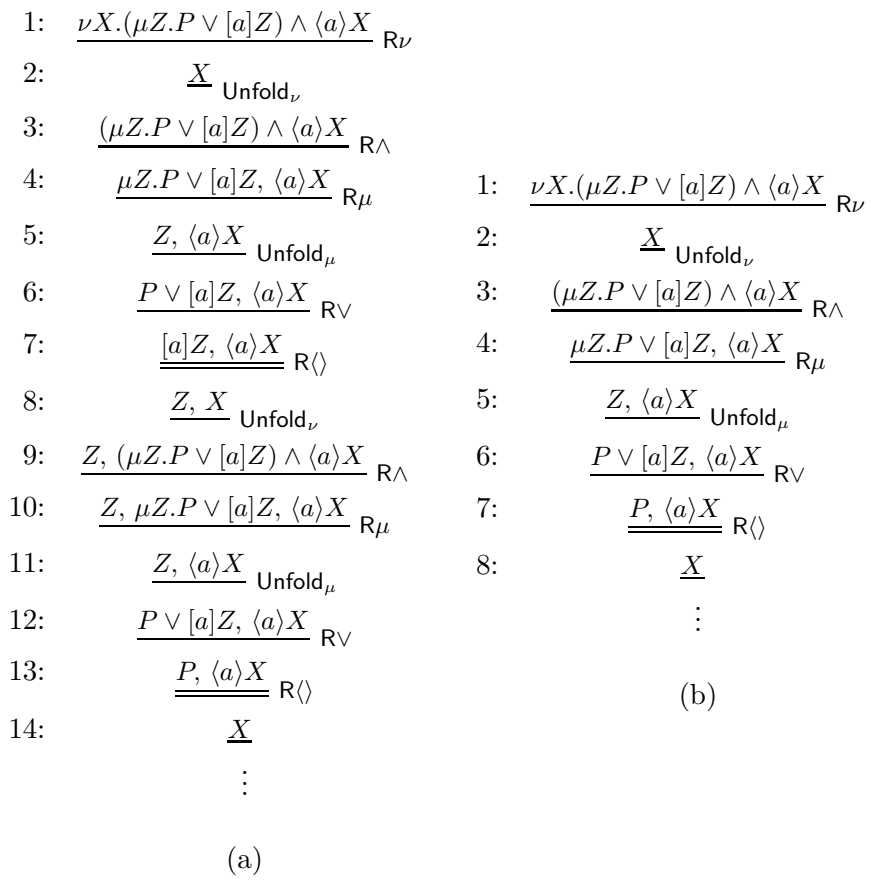


Figure 4.2: Some fragments of tableaux for $\nu X.(\mu Z.P \vee [a]Z) \wedge \langle a\rangle X$ in Example 4.2.

history of trails into each goal and uses the recorded trail information to determine whether a node is a successful terminal.

4.2 Tableau System ACON

Before we present a complete solution, let us first consider a simple tableau system, which we shall refer to as ACON. This tableau system can be seen as a simplification of the tableau system in [Koz83]. The idea is that, when the rule Rμ is applied to a μ-formula $\mu Z.\psi$ at some node u, we introduce a "name", say z, and attach it to the formula reduced from $\mu Z.\psi$ in the subgoal and then to each formula reduced from the latter formula, and so on. This propagation of name z is terminated when a variable higher than Z is reached. Thus, in essence, for each formula γ in a node v and each μ-variable Z, we are recording the highest ancestor u of v such that there is a trail from $(u, \mu Z.\psi)$ to (v, γ) which does not go through a variable higher than Z. As shown below, for a certain fragment of the logic, called *aconjunctive formulae*, it is possible to define simple termination and success conditions which guarantee soundness and completeness.

Let us explain the detail of ACON. Suppose ϕ is a closed and guarded formula. We assume a linear ordering $X_1, ..., X_n$ of all the variables in ϕ such that X_i higher than X_j implies $i < j$. For each μ-variable Z in ϕ, we assume a sequence $z^1, z^2, ...$ of distinct symbols, called *names for* Z (it is also assumed that names for different μ-variables are distinct). As we later show, the number of names required to build a tableau for ϕ is bounded by the length of ϕ. For convenience, we use small-case letters z, y, x or their scripted versions to denote names.

A goal in an ACON-tableau is a sequent of the form $\Theta \vdash \Gamma$ where

- Θ is a sequence of distinct names, called a *global sequence*,
- Γ is a set of *augmented formulae* of the form ψ^ρ, for some formula ψ and sequence $\rho = \rho(Z_1)...\rho(Z_n)$ of distinct names, where each $\rho(Z_i)$ is a *name* for Z_i and $Z_1, ..., Z_n$ $(n \geq 0)$ are some μ-variables in ϕ such that Z_i is higher than Z_{i+1} for each $i < n$.[1] It is required that each name in a sequence ρ in Γ occurs in Θ and each name in Θ occurs in some sequence ρ in Γ.

The initial goal is $\vdash \phi$ (i.e. ϕ is augmented with the empty sequence).

Definition 4.3 (Ordering of Names)**.** Names in a global sequence Θ are linearly ordered based on their positions in Θ: for any names y and z in Θ, $y <_\Theta z$ iff y occurs before z in Θ; and $y \leq_\Theta z$ iff $y <_\Theta z$ or $y = z$. This extends to sequences of names in a lexicographical manner as follows: for any sequences ρ, ρ' of names in Θ, $\rho \prec_\Theta \rho'$ iff

- for some j, $\rho(j)$ and $\rho'(j)$ are names for the *same* variable,
- $\rho(j) <_\Theta \rho'(j)$ and $\rho(i) = \rho'(i)$, for each $i < j$;

[1] Alternatively, ρ may be defined as a function whose domain is a set of μ-variables $Z_1 \prec ... \prec Z_n$ and, for each Z_i, $\rho(Z_i)$ is a name for Z_i.

Similarly, $\rho \preceq_\Theta \rho'$ iff $\rho \prec_\Theta \rho'$ or $\rho = \rho'$.

Note that the ordering $\prec_\Theta$ is *not* a total ordering. For example, suppose $\Theta = z^1 y^1 y^2$. Then $z^1 y^1 \prec_\Theta z^1 y^2$, but z^1 and $z^1 y^1$ are not comparable. Clearly, $\prec_\Theta$ is transitive and anti-symmetric.

Given a sequence ρ and a variable X, $\rho \restriction X$ denotes the sequence obtained from ρ by removing all the names for any variable appearing later than X in the sequence $X_1, ..., X_n$ assumed earlier. Similarly, for any number n, $\rho \restriction n$ denotes the sequence of the first n names in ρ. For example, suppose $\rho = x_1 x_2 x_4 x_5$ where each x_i is a name for a μ-variable X_i. Then $\rho \restriction X_3$ denotes $x_1 x_2$ and $\rho \restriction 3$ denotes $x_1 x_2 x_4$.

Tableau rules. The tableau rules are given below. In the rules Unfold_σ and Thin, Θ' denotes the result of removing the names in Θ not occurring in any augmented formula in the subgoal; and similarly for each Θ_i in the i-th subgoal of the rule $\mathsf{R}\langle\rangle$. Notice that, in the rule $\mathsf{R}\vee$, no names are removed from the goal, so the global sequence remains unchanged.

$$\mathsf{R}\wedge: \quad \frac{\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma}{\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma}$$

$$\mathsf{R}\vee: \quad \frac{\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma}{\Theta \vdash \psi_i^\rho, \Gamma} \quad i \in \{1, 2\}$$

$$\mathsf{R}\mu: \quad \frac{\Theta \vdash (\mu Z.\psi)^\rho, \Gamma}{\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma} \quad z^i \text{ is the } \textit{first} \text{ name for } Z \textit{ not} \text{ occurring in } \Theta.$$

$$\mathsf{R}\nu: \quad \frac{\Theta \vdash (\nu X.\psi)^\rho, \Gamma}{\Theta \vdash X^\rho, \Gamma}$$

$$\mathsf{Unfold}_\sigma: \quad \frac{\Theta \vdash X^\rho, \Gamma}{\Theta' \vdash \psi^{\rho \restriction X}, \Gamma} \quad X \text{ identifies } \sigma X.\psi.$$

$$\mathsf{R}\langle\rangle: \quad \frac{\Theta \vdash (\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma}{\Theta_1 \vdash \psi_1^{\rho_1}, \Gamma_{a_1} \mid ... \mid \Theta_n \vdash \psi_n^{\rho_n}, \Gamma_{a_n}} \quad n \geq 1$$

where

- Γ contains only literals and $[\cdot]$-formulae, and
- for each action a, $\Gamma_a = \{\psi^\rho \mid ([a]\psi)^\rho \in \Gamma\}$.

$$\mathsf{Thin}: \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma}$$

if $\rho \prec_\Theta \rho'$ or,
for some μ-variable Z, $\rho' \restriction Z$ is a *proper* prefix of $\rho \restriction Z$.

Remark 4.4.

(1) In the rule $\mathsf{R}\mu$, a new name for the μ-variable Z is added to the global sequence. In order to bound the number of possible goals, we always choose the first name z^i for such μ-variable (i.e. one with the least i) not occurring in Θ.

(2) If Unfold_σ is applied at node u labelled with $\Theta \vdash X^\rho, \Gamma$ (thus creating a subgoal $\Theta' \vdash \psi^{\rho[X}, \Gamma$), we say that X^ρ is *unfolded* at node u.

(3) The *thinning rule* Thin helps bounding the size of a goal by eliminating one formula in a pair of the form $\psi^\rho, \psi^{\rho'}$, where $\rho \neq \rho'$, in the goal. To ensure soundness, we must be careful in choosing which formula to discard. There are many ways to do so. The definition given here (which employs the ordering $\prec_\Theta$) is chosen because it can be reused in the tableau system TS for the full logic given in the next section. Another possible definition is to compare ρ and ρ' in the standard lexicographical manner using the global sequence Θ in the goal as the yardstick and keep the smaller one.

Although the ordering $\prec_\Theta$ is *not* total, it can be shown that, for any pair of formulae $\psi^\rho, \psi^{\rho'}$ (where $\rho \neq \rho'$) in a goal, the thinning condition adopted here uniquely chooses one of these formulae to discard. The proof is given in the next section (see Lemma 4.31).

Restriction. In order to guarantee finiteness, when constructing a tableau, the rule Thin is given the highest priority, i.e. rule Thin is always applied whenever possible.

Termination. A *terminal* is a leaf $u : \Theta \vdash \Gamma$ such that *one* of the following conditions hold:

T1. Γ contains a complementary pair of literals.
T2. Γ is a consistent set of literals and $[\cdot]$-formulae.
T3. u has a proper ancestor $v : \Theta \vdash \Gamma$, called the *companion* of u.

When constructing a tableau, if a terminal is reached, we stop applying a tableau rule to that node.

Success. It is more intuitive to define unsuccessful terminals first. An *unsuccessful terminal* is a terminal $u : \Theta \vdash \Gamma$ such that *one* of the following holds.

U1. u satisfies T1 (i.e. Γ contains a complementary pair of literals).
U2. u has a companion v such that a μ-variable Z^ρ is unfolded between v and u, and the name $\rho(Z)$ occurs throughout the path from v to u.

A terminal is said to be *successful* otherwise. Thus a terminal $u : \Theta \vdash \Gamma$ is *successful* iff *one* of the following holds.

S1. u satisfies T2 (i.e. Γ is a consistent set of literals and $[\cdot]$-formulae).

S2. u has a companion v such that, for any μ-variable Z^ρ, if Z^ρ is unfolded between v and u, then there is a goal on the path from v to u where the name $\rho(Z)$ does *not* occur.

A *successful tableau* $\mathcal{T}$ is a *finite* tableau all whose leaves are successful terminals.

A successful tableau can be seen as a model for the initial formula. In fact, in a successful tableau, if we identify each terminal which satisfies rule T3 with its companion and define the notion of trails as in tableau system TS_0, we can show that the tableau does not contain a μ-trail. It is then straightforward to extract a model from the μ-trail-free tableau and prove that it satisfies the initial formula. We shall explain this soundness proof of ACON in detail later. For now, let us look at an example of a successful tableau.

Example 4.5. Consider the formula ϕ from Example 4.1. The formula has a successful tableau in ACON as shown in Figure 4.3. By condition T3, node 20 is a terminal with node 6 as its companion; and, by condition S2, it is successful since there is no name which occurs throughout the path from node 6 to node 20.

It is interesting to compare the TS_0-tableau for this formula in Figure 4.1 with this ACON-tableau. Notice that, in the former tableau, nodes 7 and 14 have the same goal. With the annotation of names, the goals at nodes 7 and 14 in the ACON-tableau are now distinct.

The tableau system ACON is *not* generally complete. Determining whether a terminal is successful simply by checking for an unfolding of a μ-variable is too restrictive. As seen in the following example, an ACON-tableau may be declared unsuccessful even though the model corresponding to the tableau satisfies the initial formula.

Example 4.6. The formula

$$\neg P \wedge \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z)$$

is satisfiable (for example, over a linear model, the formula is true at the initial state iff every state in the model has an a-successor and P is almost always true). However, all attempts to find a successful ACON-tableau for it fail. One example of unsuccessful ACON-tableaux for this formula is given in Figure 4.4. In this tableau, node 14 is a terminal with node 9 as its companion (condition T3); and, by condition U2, it is an unsuccessful terminal because Z^{z^1} is unfolded at node 11 and name z^1 occurs in every goal between node 9 and node 14.

Notice that, although Z^{z^1} is unfolded at node 11, there is no trail from node 9 to the leaf node 14 which goes through such unfolding (this is because, in the application of rule R$\vee$ to $(P \vee \nu X.([a]X \wedge \langle a\rangle Z))^{z^1}$ at node 12, the first disjunct P is chosen, and hence when rule R$\langle\rangle$ is applied at node 13, the trail from Z^{z^1} at node 11 terminates). This means that the tableau obtained by identifying node 14 with node 9 does not

$$
\begin{array}{ll}
1: & \dfrac{\vdash (\nu X_1.(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1) \wedge (\nu X_2.(\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2)}{\vdash \nu X_1.(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ \nu X_2.(\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{R}\wedge \\
2: & \dfrac{\vdash \nu X_1.(\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ \nu X_2.(\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}{\vdash X_1,\ X_2}\ \mathsf{R}\nu \\
3: & \dfrac{\vdash X_1,\ X_2}{\vdash (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{Unfold}_\nu \\
4: & \dfrac{\vdash (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}{\vdash \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\wedge \\
5: & \dfrac{\vdash \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{z^1 \vdash Z^{z^1},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu \\
6: & \dfrac{z^1 \vdash Z^{z^1},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{z^1 y^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ [a]X_2}\ \mathsf{R}\mu \\
7: & \dfrac{z^1 y^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ [a]X_2}{z^1 y^1 \vdash (P \vee \langle a\rangle Z)^{z^1},\ [a]X_1,\ (\neg P \vee \langle a\rangle Y)^{y^1},\ [a]X_2}\ \mathsf{Unfold}_\mu \\
8: & \dfrac{z^1 y^1 \vdash (P \vee \langle a\rangle Z)^{z^1},\ [a]X_1,\ (\neg P \vee \langle a\rangle Y)^{y^1},\ [a]X_2}{\underline{\underline{z^1 y^1 \vdash P^{z^1},\ [a]X_1,\ \langle a\rangle Y^{y^1},\ [a]X_2}}}\ \mathsf{R}\vee \\
9: & \dfrac{\underline{\underline{z^1 y^1 \vdash P^{z^1},\ [a]X_1,\ \langle a\rangle Y^{y^1},\ [a]X_2}}}{y^1 \vdash X_1,\ Y^{y^1},\ X_2}\ \mathsf{R}\langle\rangle \\
10: & \dfrac{y^1 \vdash X_1,\ Y^{y^1},\ X_2}{y^1 \vdash (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ Y^{y^1},\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{Unfold}_\nu \\
11: & \dfrac{y^1 \vdash (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ Y^{y^1},\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}{y^1 \vdash \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ Y^{y^1},\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\wedge \\
12: & \dfrac{y^1 \vdash \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ Y^{y^1},\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{y^1 z^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu \\
13: & \dfrac{y^1 z^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{y^1 z^1 y^2 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ Y^{y^2},\ [a]X_2}\ \mathsf{R}\mu \\
14': & \dfrac{y^1 z^1 y^2 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ Y^{y^2},\ [a]X_2}{y^1 z^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ [a]X_2}\ \mathsf{Thin} \\
14: & \dfrac{y^1 z^1 \vdash Z^{z^1},\ [a]X_1,\ Y^{y^1},\ [a]X_2}{y^1 z^1 \vdash (P \vee \langle a\rangle Z)^{z^1},\ [a]X_1,\ (\neg P \vee \langle a\rangle Y)^{y^1},\ [a]X_2}\ \mathsf{Unfold}_\mu \\
15: & \dfrac{y^1 z^1 \vdash (P \vee \langle a\rangle Z)^{z^1},\ [a]X_1,\ (\neg P \vee \langle a\rangle Y)^{y^1},\ [a]X_2}{\underline{\underline{y^1 z^1 \vdash \langle a\rangle Z^{z^1},\ [a]X_1,\ \neg P^{y^1},\ [a]X_2}}}\ \mathsf{R}\vee \\
16: & \dfrac{\underline{\underline{y^1 z^1 \vdash \langle a\rangle Z^{z^1},\ [a]X_1,\ \neg P^{y^1},\ [a]X_2}}}{z^1 \vdash Z^{z^1},\ X_1,\ X_2}\ \mathsf{R}\langle\rangle \\
17: & \dfrac{z^1 \vdash Z^{z^1},\ X_1,\ X_2}{z^1 \vdash Z^{z^1},\ (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}\ \mathsf{Unfold}_\nu \\
18: & \dfrac{z^1 \vdash Z^{z^1},\ (\mu Z.P \vee \langle a\rangle Z) \wedge [a]X_1,\ (\mu Y.\neg P \vee \langle a\rangle Y) \wedge [a]X_2}{z^1 \vdash Z^{z^1},\ \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\wedge \\
19: & \dfrac{z^1 \vdash Z^{z^1},\ \mu Z.P \vee \langle a\rangle Z,\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{z^1 z^2 \vdash Z^{z^1},\ Z^{z^2},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{R}\mu \\
20': & \dfrac{z^1 z^2 \vdash Z^{z^1},\ Z^{z^2},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{z^1 \vdash Z^{z^1},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}\ \mathsf{Thin} \\
20: & \dfrac{z^1 \vdash Z^{z^1},\ [a]X_1,\ \mu Y.\neg P \vee \langle a\rangle Y,\ [a]X_2}{\text{SUCCESSFUL}}
\end{array}
$$

Figure 4.3: A successful ACON-tableau for the formula ϕ in Example 4.1. By condition T3, node 20 is a terminal with node 6 as its companion. Since there is no name which occurs in every node between node 6 and node 20, by condition S2, the terminal node 20 is successful.

contain a μ-trail. As we later show, this implies that the model corresponding to the tableau satisfies the given formula.

$$
\begin{array}{ll}
1: & \dfrac{\vdash \neg P \wedge \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z)}{} \\
2: & \dfrac{\vdash \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z), \neg P}{} \ \mathsf{R}\mu \\
3: & \dfrac{z^1 \vdash Z^{z^1}, \neg P}{} \ \mathsf{Unfold}_\mu \\
4: & \dfrac{z^1 \vdash (P \vee \nu X.([a]X \wedge \langle a\rangle Z))^{z^1}, \neg P}{} \ \mathsf{R}\vee \\
5: & \dfrac{z^1 \vdash \nu X.([a]X \wedge \langle a\rangle Z)^{z^1}, \neg P}{} \ \mathsf{R}\nu \\
6: & \dfrac{z^1 \vdash X^{z^1}, \neg P}{} \ \mathsf{Unfold}_\nu \\
7: & \dfrac{z^1 \vdash ([a]X \wedge \langle a\rangle Z)^{z^1}, \neg P}{} \ \mathsf{R}\wedge \\
8: & \dfrac{\!\!\dfrac{}{}\!\!}{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, \neg P} \ \mathsf{R}\langle\rangle \\
9: & \dfrac{z^1 \vdash X^{z^1}, Z^{z^1}}{} \ \mathsf{Unfold}_\nu \\
10: & \dfrac{z^1 \vdash ([a]X \wedge \langle a\rangle Z)^{z^1}, Z^{z^1}}{} \ \mathsf{R}\wedge \\
11: & \dfrac{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, Z^{z^1}}{} \ \mathsf{Unfold}_\mu \\
12: & \dfrac{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, (P \vee \nu X.([a]X \wedge \langle a\rangle Z))^{z^1}}{} \ \mathsf{R}\vee \\
13: & \dfrac{\!\!\dfrac{}{}\!\!}{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, P^{z^1}} \ \mathsf{R}\langle\rangle \\
14: & \dfrac{z^1 \vdash X^{z^1}, Z^{z^1}}{} \\
 & \text{UNSUCCESSFUL}
\end{array}
$$

Figure 4.4: An unsuccessful ACON-tableau for Example 4.6. By condition T3, node 14 is a terminal with node 9 as its companion. Since Z^{z^1} is unfolded at node 11 and the name z^1 occurs in every node between node 9 and node 14, by condition U2, node 14 is an unsuccessful terminal.

The previous example raises a question of whether there is a more generous success condition for ACON which captures precisely the tableaux which are free of μ-trails (such a success condition should, therefore, classify the tableau in the previous example as successful). Unfortunately, this is not possible unless the termination condition T3 (which requires that a branch is terminated whenever there is a repeating goal) is relaxed. As seen from the example below, there is a satisfiable formula in which every potential tableau is prematurely terminated before reaching a tableau which is free of μ-trails.

Example 4.7. Let ϕ be the conjunction of the following formulae

$$\neg Q$$
$$\mu Z.(P \vee \langle a\rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)$$
$$\mu Y.(\neg P \vee \langle a\rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)$$

ϕ is satisfiable yet it has no successful tableau in ACON. An unsuccessful ACON-tableau

for ϕ is shown in Figure 4.5. By condition T3, Node 19 is a terminal with node 11 as its companion; and, by condition U2, it is an unsuccessful terminal because Y^{y^1} is unfolded at node 12 and the name y^1 occurs in every goal between node 19 and its companion.

Notice that this tableau contains a μ-trail (assuming that we identify node 19 with its companion, node 11):

$$...\rightarrow(11,Y^{y^1})\rightarrow(12,Y^{y^1})\rightarrow(13,(\neg P\vee\langle a\rangle Y)\wedge(Q\vee\nu X_2.[a]Y\wedge[a]X_2)^{y^1})\rightarrow(14,(\neg P\vee\langle a\rangle Y)^{y^1})\rightarrow$$
$$(15,\langle a\rangle Y^{y^1})\rightarrow(16,\langle a\rangle Y^{y^1})\rightarrow(17,\langle a\rangle Y^{y^1})\rightarrow(18,\langle a\rangle Y^{y^1})\rightarrow(19,Y^{y^1})\rightarrow(11,Y^{y^1})\rightarrow...$$

In fact, the model corresponding to this tableau does not satisfy ϕ. In the next section (Example 4.35), we will show that this tableau can be extended into a tableau whose corresponding model satisfies ϕ.

1: $\dfrac{\vdash\neg Q,\mu Z.(P\vee\langle a\rangle Z)\wedge(Q\vee\nu X_1.[a]Z\wedge[a]X_1),\mu Y.(\neg P\vee\langle a\rangle Y)\wedge(Q\vee\nu X_2.[a]Y\wedge[a]X_2)}{}\ \mathsf{R}\mu$

2: $\dfrac{z^1\vdash\neg Q,Z^{z^1},\mu Y.(\neg P\vee\langle a\rangle Y)\wedge(Q\vee\nu X_2.[a]Y\wedge[a]X_2)}{}\ \mathsf{R}\mu$

3: $\dfrac{z^1y^1\vdash\neg Q,Z^{z^1},Y^{y^1}}{}\ \mathsf{Unfold}_\mu$

4: $\dfrac{z^1y^1\vdash\neg Q,(P\vee\langle a\rangle Z)\wedge(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},Y^{y^1}}{}\ \mathsf{Unfold}_\mu$

5: $\dfrac{z^1y^1\vdash\neg Q,(P\vee\langle a\rangle Z)\wedge(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},(\neg P\vee\langle a\rangle Y)\wedge(Q\vee\nu X_2.[a]Y\wedge[a]X_2)^{y^1}}{}\ \mathsf{R}\wedge$

6: $\dfrac{z^1y^1\vdash\neg Q,(P\vee\langle a\rangle Z)^{z^1},(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},(\neg P\vee\langle a\rangle Y)^{y^1},(Q\vee\nu X_2.[a]Y\wedge[a]X_2)^{y^1}}{}\ \mathsf{R}\vee$

7: $\dfrac{z^1y^1\vdash\neg Q,P^{z^1},\nu X_1.[a]Z\wedge[a]X_1^{z^1},\langle a\rangle Y^{y^1},\nu X_2.[a]Y\wedge[a]X_2^{y^1}}{}\ \mathsf{R}\nu$

8: $\dfrac{z^1y^1\vdash\neg Q,P^{z^1},X_1^{z^1},\langle a\rangle Y^{y^1},X_2^{y^1}}{}\ \mathsf{Unfold}_\nu$

9: $\dfrac{z^1y^1\vdash\neg Q,P^{z^1},([a]Z\wedge[a]X_1)^{z^1},\langle a\rangle Y^{y^1},([a]Y\wedge[a]X_2)^{y^1}}{}\ \mathsf{R}\wedge$

10: $\dfrac{\!\!\dfrac{z^1y^1\vdash\neg Q,P^{z^1},[a]Z^{z^1},[a]X_1^{z^1},\langle a\rangle Y^{y^1},[a]Y^{y^1},[a]X_2^{y^1}}{}\ \mathsf{R}\langle\rangle}{}$

11: $\dfrac{z^1y^1\vdash Z^{z^1},X_1^{z^1},Y^{y^1},X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

12: $\dfrac{z^1y^1\vdash(P\vee\langle a\rangle Z)\wedge(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},X_1^{z^1},Y^{y^1},X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

13: $\dfrac{z^1y^1\vdash(P\vee\langle a\rangle Z)\wedge(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},X_1^{z^1},(\neg P\vee\langle a\rangle Y)\wedge(Q\vee\nu X_2.[a]Y\wedge[a]X_2)^{y^1},X_2^{y^1}}{}\ \mathsf{R}\wedge$

14: $\dfrac{z^1y^1\vdash(P\vee\langle a\rangle Z)^{z^1},(Q\vee\nu X_1.[a]Z\wedge[a]X_1)^{z^1},X_1^{z^1},(\neg P\vee\langle a\rangle Y)^{y^1},(Q\vee\nu X_2.[a]Y\wedge[a]X_2)^{y^1},X_2^{y^1}}{}\ \mathsf{R}\vee$

15: $\dfrac{z^1y^1\vdash P^{z^1},Q^{z^1},X_1^{z^1},\langle a\rangle Y^{y^1},Q^{y^1},X_2^{y^1}}{}\ \mathsf{Thin}$

16: $\dfrac{z^1y^1\vdash P^{z^1},Q^{z^1},X_1^{z^1},\langle a\rangle Y^{y^1},X_2^{y^1}}{}\ \mathsf{Unfold}_\nu$

17: $\dfrac{z^1y^1\vdash P^{z^1},Q^{z^1},([a]Z\wedge[a]X_1)^{z^1},\langle a\rangle Y^{y^1},([a]Y\wedge[a]X_2)^{y^1}}{}\ \mathsf{R}\wedge$

18: $\dfrac{\!\!\dfrac{z^1y^1\vdash P^{z^1},Q^{z^1},[a]Z^{z^1},[a]X_1^{z^1},\langle a\rangle Y^{y^1},[a]Y^{y^1},[a]X_2^{y^1}}{}\ \mathsf{R}\langle\rangle}{}$

19: $\dfrac{z^1y^1\vdash Z^{z^1},X_1^{z^1},Y^{y^1},X_2^{y^1}}{\text{UNSUCCESSFUL}}$

Figure 4.5: An unsuccessful ACON-tableau for the satisfiable formula ϕ in Example 4.7. By condition T3, node 19 is a terminal with node 11 as its companion. Since Y^{y^1} is unfolded at node 12 and the name y^1 occurs in every node between node 11 and node 19, by condition U2, the terminal node 19 is unsuccessful.

There is a fragment of the modal μ-calculus for which ACON is sound and complete. This fragment of the logic, called *aconjunctive formulae* (hence the name ACON for the tableau system), was first identified by Kozen [Koz83].

Definition 4.8 (Aconjunctive Formulae)**.** A formula ϕ (in positive normal form) is said to be *aconjunctive in variable* X iff for any subformula $\psi_1 \wedge \psi_2$ of ϕ, X is active in *at most* one ψ_i. ϕ is then said to be *aconjunctive* iff it is aconjunctive in every μ-variable in it.

For example, the formulae $\mu X.\langle a\rangle X \vee \langle b\rangle X$ and $\nu X.(\mu Z.\nu Y.[a]Z \vee [a]Y) \wedge \langle a\rangle X$ are aconjunctive, whereas the formulae $\mu X.\langle a\rangle X \wedge \langle b\rangle X$ and $\nu X.(\mu Z.\nu Y.[a]Z \wedge [a]Y) \wedge \langle a\rangle X$ are not (in the latter formula, Z is active in both $[a]Z$ and $[a]Y$).

It is obvious that the class of aconjunctive formulae is closed under subformulae. A tableau for an aconjunctive formula has the following property which is required in the completeness proof.

Lemma 4.9. *In any ACON-tableau for an* aconjunctive *formula, if a goal contains formulae* $\psi_1^{\rho_1}, \psi_2^{\rho_2}$ *where* $\psi_1 \neq \psi_2$ *and a* μ*-variable* Z *is active in both* ψ_1 *and* ψ_2*, then* $\rho_1(Z) \neq \rho_2(Z)$*.*

Proof. This property can be shown as an invariant when constructing a tableau for an aconjunctive formula ϕ. The initial goal obviously has this property. Suppose the goal $\Theta \vdash \Gamma$ at a node u has this property. By considering the rule applied at u, it can be shown that every subgoal of u must have this property. We explain the non-obvious cases below:

- rule R$\wedge$ is applied at u, $\Gamma = (\psi_1 \wedge \psi_2)^\rho, \Gamma'$, and the subgoal is $\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma'$. Since ϕ is aconjunctive, every μ-variable is active in at most one formula in $\{\psi_1, \psi_2\}$. Since every variable active in ψ_1 or ψ_2 is active in $\psi_1 \wedge \psi_2$, it follows that the above property still holds for the subgoal.
- rule Rμ is applied at u, $\Gamma = (\mu Y.\psi)^\rho, \Gamma'$, and the subgoal is $\Theta \cdot y \vdash Y^{\rho \cdot y}, \Gamma'$ for some name y not occurring in Θ. Suppose the subgoal does not have the above property. Since the original goal has the property, this can happen only when there is a formula $\psi^{\rho'} \in \Gamma'$ ($\psi^{\rho'} \neq Y^{\rho \cdot y}$) such that a μ-variable Z is active in both Y and ψ, and $\rho'(Z) = (\rho \cdot y)(Z)$. Clearly $Z \neq Y$ because $(\rho \cdot y)(Y) = y$ does not occur in Γ'. So Z must be higher than Y and hence also active in $\mu Y.\psi$. But this implies that the original goal fails the above property. Therefore, the property still holds for the subgoal.

□

We now turn to prove the soundness and completeness of ACON with respect to the aconjunctive fragment. Note that much of the terminology in the following proofs will be reused, some in a more general form, for the tableau system TS in the next section.

Finiteness. Previously we required that a successful tableau must be finite. Indeed, it can be shown that every ACON-tableau is *finite*. This follows from the restriction that rule Thin is applied whenever possible and from the canonical choice of a new name introduced by rule Rμ. These two conditions ensure that there are finitely many possible goals in a tableau.

Lemma 4.10. *For each μ-variable Z, the names for Z occurring in each goal in any ACON-tableau are among $z^1, ..., z^{|\phi|}$.*

Proof. There can be at most one name for each μ-variable in each formula. Since rule Thin is always applied whenever possible, there can be at most $|\phi|$ formulae in the goal when applying rule Rμ. And since Rμ adds the first unused name for the μ-variable being unfolded, the names for each μ-variable Z in a goal must be among $z^1, ..., z^{|\phi|}$. □

Lemma 4.11. *Every ACON-tableau for ϕ is a finite tree of degree $O(|\phi|)$ and height $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$.*

Proof. Let n denote the length of ϕ, and suppose there are m μ-variables in ϕ, namely, $Z_1, ..., Z_m$. In the proof below, by a *goal*, we mean a goal in any ACON-tableau for ϕ.

We begin by showing that there is a bound on the number of possible goals where the rule Thin is *not* applicable. We shall first count the number of distinct sets Γ such that $\Theta \vdash \Gamma$, for some Θ, is such a goal. Let $\mathcal{C}$ be the collection of all such sets Γ. Hence, each Γ in $\mathcal{C}$ has the following properties:

(1) For any subformula ψ of ϕ, there is *at most* one formula of the form ψ^ρ in Γ.
(2) For each μ-variable Z in ϕ, the names for Z in Γ are among the first n ones, i.e. $z^1, ..., z^n$.

(1) follows from the fact that Γ belongs to a goal where Thin is not applicable. (2) follows from the previous lemma.

Recall that, for any formula ψ^ρ, ρ contains *at most* one name for each μ-variable. We shall then represent each set Γ in $\mathcal{C}$ as a *partial function*

$$f_\Gamma : \mathrm{Sub}(\phi) \rightharpoonup \prod_{i=1}^{m} (\mathsf{Name}_n(Z_i) \cup \{nil\}),$$

where $\mathsf{Name}_n(Z_i)$ is the set of the first n names for Z_i and "*nil*" is some symbol distinct from every name. Namely, for each $\Gamma \in \mathcal{C}$, f_Γ is given as follows: for each $\psi \in \mathrm{Sub}(\phi)$,

- if there is a (unique) sequence ρ such that $\psi^\rho \in \Gamma$, then $f_\Gamma(\psi) = (z_1, ..., z_m)$ where, for each $i \leq m$,
  - z_i is the name for Z_i in ρ, if such a name exists, and
  - $z_i = nil$, if there is no name for Z_i in ρ;
- if there is no formula of the form ψ^ρ in Γ, then $f_\Gamma(\psi)$ is undefined.

Clearly, this representation provides a one-one mapping from $\mathcal{C}$ into the set of all such partial functions (i.e. for any distinct Γ_1, Γ_2 in $\mathcal{C}$, $f_{\Gamma_1} \neq f_{\Gamma_2}$). Hence, $|\mathcal{C}|$ is bounded by the number of such partial functions, which is no greater than $((n+1)^m + 1)^n$.

Next, we take into account the ordering of names. Notice that, by the previous lemma, the number of names occurring in each set $\Gamma \in \mathcal{C}$ is no greater than mn. Hence, the names in each set Γ can be linearly ordered in at most $(mn)!$ ways. Therefore, the number of possible goals where the rule Thin is *not* applicable is $\leq (mn)! \cdot ((n+1)^m+1)^n$.

Observe that, when constructing a tableau, we never need to apply the rule Thin consecutively more than twice. Recall that Thin is the first rule to be applied to a goal which contains a *redundant pair*, i.e. a pair of the form $\psi^\rho, \psi^{\rho'}$ where $\rho \neq \rho'$. A tableau rule other than $\mathsf{R}\wedge$ when applied to a goal which does not contain any redundant pair can create at most one redundant pair in each subgoal. The rule $\mathsf{R}\wedge$ may create one of the following types of redundancies in the subgoal:

- a redundant triple ψ^{ρ_1}, ψ^{ρ_2}, ψ^{ρ_3}, where ρ_1, ρ_2, ρ_3 are distinct, or
- two redundant pairs $\psi^{\rho_1}, \psi^{\rho_2}$ and $\psi'^{\rho_3}, \psi'^{\rho_4}$, where $\psi \neq \psi'$, $\rho_1 \neq \rho_2$, and $\rho_3 \neq \rho_4$, or
- just one redundant pair, ψ^ρ, $\psi^{\rho'}$, where $\rho \neq \rho'$.

Clearly, these redundancies can be eliminated by applying Thin once or twice in a row. This means that a branch longer than $3 \cdot (mn)! \cdot ((n+1)^m + 1)^n$ must contain more than $(mn)! \cdot ((n+1)^m + 1)^n$ nodes which are labelled by the goals in which Thin is *not* applicable. By what we have shown above, such a branch must contain a repeat of a goal. From the termination condition T3 and the restriction that a terminal node is not further expanded, the height of any tableau for ϕ is therefore bounded by $3 \cdot (mn)! \cdot ((n+1)^m + 1)^n + 1 = 2^{O(mn\log(mn))}$ (which equals to $2^{O(mn\log(n))}$ because $m < n$).

The degree of a tableau for ϕ cannot exceed the number of $\langle\cdot\rangle$-subformulae of ϕ, and is hence bounded by $O(n)$. □

4.2.1 Soundness

Suppose $\mathcal{T}$ is a successful ACON-tableau for a guarded and closed formula ϕ. $\mathcal{T}$ can be seen as a tree-with-backedges structure (where the backedges are from the leaves to their companions). As in the soundness proof of tableau system TS_0, a model for ϕ can be constructed by identifying each "modal node" as a state. Here, a *modal node* is either a node where rule $\mathsf{R}\langle\rangle$ is applied or a leaf node which has no companion (i.e. a leaf node which contains only $[\cdot]$-formulae and literals). For convenience, we use the letters s, t and their scripted versions to denote modal nodes.

To define the transition relation, we need some extra notation.

Definition 4.12. Suppose $\mathcal{T}$ is a tableau in ACON. For any nodes u, v in $\mathcal{T}$, we write $u \Rightarrow v$ when either v is a child of u or u is a leaf and v is its companion.

For each modal node s, define the set

$[s] = \{u \mid$ there is a path $u = u_1 \Rightarrow ... \Rightarrow u_n = s$ $(n \geq 1)$ such that $\mathsf{R}\langle\rangle$ is *not* applied at u_i for each $i < n$ $\}$.

Clearly, for any distinct modal nodes s, t, the sets $[s]$ and $[t]$ must be disjoint. It thus follows from the following lemma that for each node u there exists a *unique* modal node s such that $u \in [s]$.

Lemma 4.13. *Suppose $\mathcal{T}$ is a tableau for a* guarded *formula. For each node u in $\mathcal{T}$, there is a path $u = u_1 \Rightarrow ... \Rightarrow u_n$ $(n \geq 1)$ such that u_n is a modal node.*

Proof. If this is not the case, there will be an infinite sequence $u_1 \Rightarrow u_2 \Rightarrow ...$ where rule $\mathsf{R}\langle\rangle$ is not applied at u_i for each $i \geq 1$. This is clearly impossible if the formula ϕ is guarded. □

Definition 4.14. Suppose $\mathcal{T}$ is a tableau for a *guarded* formula. Define the *model corresponding to* $\mathcal{T}$ to be $\mathcal{M}_\mathcal{T} = \langle S, \{R_a\}_{a \in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}} \rangle$ where

- S contains all modal nodes of $\mathcal{T}$,
- sR_at iff, for some node $u \in [t]$, a formula $\langle a \rangle \psi$ in s is reduced to ψ in u (by rule $\mathsf{R}\langle\rangle$), and
- $\mathcal{V}_{\mathrm{Prop}}(P) = \{s \mid P^\rho$, for some ρ, is in the goal at s $\}$.

It can be shown that $\mathcal{M}_\mathcal{T}$ is indeed a model for ϕ, provided that $\mathcal{T}$ is successful. A simple, but somewhat indirect, way to show this is to first introduce the notion of trails on the tableau $\mathcal{T}$ as done for a TS_0-tableau, and then show that there is no μ-trail in $\mathcal{T}$. It then follows that $\mathcal{M}_\mathcal{T}$ is a model for ϕ.

We first describe the definition of *dependency relations* and *trails* on ACON-tableaux. This is very similar to such notions on TS_0-tableaux. The additional case for rule Thin is straightforward.

Definition 4.15 (Trails)**.** The *dependency relation* $\rightarrow$ on a ACON-tableau $\mathcal{T}$ is the smallest (binary) relation over pairs (u, ψ^ρ), where u is a node and ψ^ρ is in the goal at u, satisfying the following:

(a) For each node u where rule $\mathsf{R}\wedge$, $\mathsf{R}\vee$, $\mathsf{R}\mu$, $\mathsf{R}\nu$, or Unfold_σ is applied, if the formula ψ^ρ in u is reduced to $\psi'^{\rho'}$ in the child u', then $(u, \psi^\rho) \rightarrow (u', \psi'^{\rho'})$.

(b) For each node u where rule Thin is applied, if the formulae $\psi^\rho, \psi^{\rho'}$ are reduced to ψ^ρ in the child u' (i.e. $\psi^{\rho'}$ is discarded), then $(u, \psi^\rho) \rightarrow (u', \psi^\rho)$ and $(u, \psi^{\rho'}) \rightarrow (u', \psi^\rho)$.

(c) In the above cases, if a formula γ^ρ in u is *not* reduced by the rule (hence γ^ρ is also in the child u'), then $(u, \gamma^\rho) \rightarrow (u', \gamma^\rho)$.

(d) For each node u where rule $\mathsf{R}\langle\rangle$ is applied and $(\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma$ are the formulae in u, if the formula $(\langle a_i \rangle \psi_i)^{\rho_i}$ in u is reduced to $\psi_i^{\rho_i}$ in a child u_i, then $(u, (\langle a_i \rangle \psi_i)^{\rho_i}) \rightarrow (u_i, \psi_i^{\rho_i})$ and, for each formula $([a_i]\psi)^\rho \in \Gamma$, $(u, ([a_i]\psi)^\rho) \rightarrow (u_i, \psi^\rho)$.

(e) Lastly, for any leaf u which has a companion v, $(u, \psi^\rho) \to (v, \psi^\rho)$ for each ψ^ρ in u.

A *trail* in tableau $\mathcal{T}$ is a path over its dependency relation.

Terminology for trails in pre-models can be given here in the obvious way. In particular, a μ-trail (ν-trail) is an infinite trail in which the outermost variable unfolded infinitely often is a μ-variable (respectively ν-variable).

The following observation is quite clear from the tableau rules.

Observation 4.16. *Suppose ψ^ρ is an augmented formula in a goal.*

- *If ρ contains a name for a μ-variable Z, then it contains a name for each μ-variable higher than Z.*
- *If a variable X is active in ψ, then ρ contains a name for each μ-variable higher than or equal to X.*

Lemma 4.17. *Suppose $u : \Theta \vdash \Gamma$ and $u' : \Theta' \vdash \Gamma'$ are nodes in an ACON-tableau $\mathcal{T}$. If $(u, \psi^\rho) \to (u', \psi'^{\rho'})$ and a μ-variable Z is* active *in both ψ and ψ' then $\rho \restriction Z \succeq_\Theta \rho' \restriction Z$.*

Proof. Suppose $(u, \psi^\rho) \to (u', \psi'^{\rho'})$ and μ-variable Z is *active* in both ψ and ψ'. Consider the tableau rule applied at u. If the rule is other than rule Thin, then clearly $\rho \restriction Z = \rho' \restriction Z$. If Thin is applied at u, then either $\rho \succ_\Theta \rho'$ or $\rho \restriction Z'$ is a proper prefix of $\rho' \restriction Z'$ for some μ-variable Z'. Suppose the latter is the case. Hence $\rho \restriction Z' = z_1 ... z_m$ and $\rho' \restriction Z' = z_1 ... z_m z_{m+1} ... z_n$ where $0 \leq m < n$ and each z_i $(1 \leq i \leq n)$ is a name for some μ-variable which is higher than or equal to Z'. If z_{m+1} is a name for Z_{m+1}, then obviously ρ does not contain a name for Z_{m+1}. Since Z is active in ψ, by Observation 4.16, ρ must contain a name for each μ-variable higher than or equal to Z. This means that a name in ρ for a μ-variable higher than or equal to Z must be among in $z_1, ..., z_m$ (if such a name occurs after z_m, Z_{m+1} must be higher than or equal to Z, and hence ρ must contain a name for Z_{m+1} as well). This implies that $\rho \restriction Z = \rho' \restriction Z$. We may then conclude that $\rho \restriction Z \succeq_\Theta \rho' \restriction Z$. □

Lemma 4.18. *Every successful ACON-tableau does not contain a μ-trail.*

Proof. Suppose $\mathcal{T}$ is a successful tableau which contains a μ-trail. Such a μ-trail must contain a subtrail

$$(u_1, \psi_1^{\rho_1}) \to (u_2, \psi_2^{\rho_2}) \to ...$$

where a μ-variable Z is active and unfolded infinitely often. By the previous lemma,

$$\rho_1 \restriction Z \succeq_{\Theta_1} \rho_2 \restriction Z \succeq_{\Theta_2} ...$$

Since there are finitely many augmented formulae in the tableau, there must be some $j \geq 1$ such that $\psi_j^{\rho_j}$ occurs infinitely often in the above trail. This implies that $\rho_j \restriction Z =$

$\rho_{j+1} \restriction Z = ...$ Since, for infinitely many $i \geq j$, $\psi_i = Z$, it follows that $\mathcal{T}$ must contain a path $v, ..., u$, where u is a leaf and v is its companion such that Z^{ρ_i}, for some $i \geq j$, is unfolded between u and v and the name $\rho_i(Z)$ occurs throughout the path. This contradicts the assumption that each leaf of $\mathcal{T}$ is successful. Hence $\mathcal{T}$ cannot contain a μ-trail. □

Lemma 4.19. *If an ACON-tableau $\mathcal{T}$ for ϕ does not contain a μ-trail, then $\mathcal{M}_T$ is a model of ϕ.*

Proof. This is essentially the Fundamental Semantic Theorem for tableaux. The proof is similar to Lemma 4.46 in the next section. □

Theorem 4.20 (Soundness of ACON)**.** *Every guarded formula which has a successful ACON-tableau has a model where the number of states is linear in the number of nodes in the tableau.*

Proof. Suppose $\mathcal{T}$ is a successful ACON-tableau for the given (guarded and closed) formula ϕ. By Lemma 4.18, there is no μ-trail in $\mathcal{T}$. By Lemma 4.19, $\mathcal{M}_\mathcal{T}$ is a model for ϕ. The model $\mathcal{M}_\mathcal{T}$ contains the modal nodes of $\mathcal{T}$ as its states; hence the size of $\mathcal{M}_\mathcal{T}$ is clearly linear in the number of nodes in $\mathcal{T}$. □

4.2.2 Completeness

Every satisfiable *aconjunctive* formula ϕ has a successful ACON-tableau. The idea of the proof is to construct a successful tableau for ϕ by making choices which minimise a certain *measure* associated with the goals. The crucial part is to carefully define such a measure so that the constructed tableau guarantees to be successful. Informally, the measure which we are defining is a generalised form of signatures. It assigns an ordinal to each name.

Definition 4.21 (Name Signatures)**.** A *name signature* is a function $\sigma : \mathsf{Name} \to \mathbb{O}$, where Names is the set of all names.

Definition 4.22 (Ordering of Name Signatures)**.** Name signatures will be ordered with respect to a global sequence Θ in a lexicographical manner as follows. For any global sequence $\Theta = z_1...z_n$, we define the following:

- $\sigma \approx_\Theta \sigma'$ iff $\sigma(z_i) = \sigma'(z_i)$ for each i.
- $\sigma \prec_\Theta \sigma'$ iff $\sigma(z_j) < \sigma'(z_j)$ for some j and $\sigma(z_i) = \sigma'(z_i)$ for each $i < j$.
- $\sigma \preceq_\Theta \sigma'$ iff $\sigma \prec_\Theta \sigma'$ or $\sigma \approx_\Theta \sigma'$.

Clearly, $\preceq_\Theta$ is reflexive and transitive but might not be anti-symmetric. This is because, generally, not all possible names occur in Θ.

Given a name signature σ and a formula ψ^ρ, we can assign a signature for the μ-variables active in ψ based on the names in ρ and the ordinals for such names given by σ. Precisely, given a name signature σ and a sequence ρ, we define the signature σ_ρ associated with ρ as follows: for each μ-variable Z,

- $\sigma_\rho(Z) = \sigma(z)$, if ρ contains a (unique) name z for Z,
- otherwise $\sigma_\rho(Z)$ can be assigned to any chosen value.

Definition 4.23. A name signature σ is *good* for Γ iff there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \psi$, for each $\psi^\rho \in \Gamma$.

Lemma 4.24. *For any goal $\Theta \vdash \Gamma$, if Γ is satisfiable, then there is a* good *name signature for Γ.*

Proof. If Γ is satisfiable (and finite), there must be a model $\mathcal{M}$, a state s, and a least ordinal α such that $\mathcal{M}, s \models_{\langle\alpha,\ldots,\alpha\rangle} \psi$ for each formula ψ in Γ. This means that the name signature σ where $\sigma(Z) = \alpha$ for each μ-variable Z is good for Γ. □

Definition 4.25 (Signature of a Goal)**.** For each goal $\Theta \vdash \Gamma$ where Γ is satisfiable, define $\mathrm{Sig}(\Theta \vdash \Gamma)$, called the *signature of goal* $\Theta \vdash \Gamma$, to be the name signature σ such that

- σ is good for Γ,
- $\sigma \preceq_\Theta \sigma'$ for any good name signature σ' for Γ, and
- $\sigma(z) = 0$ for each name z *not* occurring in Θ.

It is obvious that a name signature σ satisfying these conditions must be unique. The existence of such a name signature follows from Lemma 4.24.

This notion of the signature of a goal is the measure which we later use in constructing a successful tableau. We first study some key properties of these signatures. The following obvious property of $\preceq_\Theta$ will be used without explicit mention.

Fact 4.26. *If Θ' is a prefix of Θ then*

- *$\sigma \preceq_\Theta \sigma'$ implies $\sigma \preceq_{\Theta'} \sigma'$;*
- *$\sigma \prec_{\Theta'} \sigma'$ implies $\sigma \prec_\Theta \sigma'$.*

Lemma 4.27. *Below Θ' denotes the result of removing all the names in Θ not occurring in any formula in the goal on the right hand side.*

(a) $\Gamma' \subseteq \Gamma$ *implies* $\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \Gamma')$.

(b) $\mathrm{Sig}(\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma)$.

(c) $\mathrm{Sig}(\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_i^\rho, \Gamma)$ *for some* $i \in \{1, 2\}$.

(d) $\mathrm{Sig}(\Theta \vdash (\mu Z.\psi)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma)$ *where z^i is a name for Z* not *occurring in Θ.*

(e) $\mathrm{Sig}(\Theta \vdash (\nu X.\psi)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash X^\rho, \Gamma)$.

(f) $\mathrm{Sig}(\Theta \vdash X^\rho, \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^{\rho\lfloor X}, \Gamma)$ *where X identifies $\nu X.\psi$.*

(g) $\mathrm{Sig}(\Theta \vdash (\langle a\rangle\psi)^\rho, \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^\rho, \Gamma_a)$ *where* $\Gamma_a = \{\gamma^{\rho'} \mid ([a]\gamma)^{\rho'} \in \Gamma\}$.

Proof. We explain (b), (d), and (f) only. Other cases are similar.

(b) Let σ be $\mathrm{Sig}(\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma)$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \psi_1 \wedge \psi_2$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. This implies that $\mathcal{M}, s \models_{\sigma_\rho} \psi_i$ for each i. Hence σ is good for $\{\psi_1^\rho, \psi_2^\rho\} \cup \Gamma$, which implies that $\sigma \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma)$.

(d) Let σ be $\mathrm{Sig}(\Theta \vdash (\mu Z.\psi)^\rho, \Gamma)$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \mu Z.\psi$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. Hence, for some ordinal α, $\mathcal{M}, s \models_{\sigma_\rho} \mu^\alpha Z.\psi$. Suppose z^i is a name for Z not occurring Θ. Let σ' be $\sigma[z^i := \alpha]$. It is easily seen that

$$\mathcal{M}, s \models_{\sigma'_{\rho \cdot z^i}} Z \text{ and } \mathcal{M}, s \models_{\sigma'_{\rho'}} \gamma,$$

for each $\gamma^{\rho'} \in \Gamma$. Therefore, σ' is a good name signature for $\{Z^{\rho \cdot z^i}\} \cup \Gamma$, and hence

$$\sigma' \quad \succeq_{\Theta \cdot z^i} \quad \mathrm{Sig}(\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma)$$

Since $\sigma \approx_\Theta \sigma'$, we have

$$\sigma \quad \succeq_\Theta \quad \mathrm{Sig}(\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma).$$

(f) Let σ be $\mathrm{Sig}(\Theta \vdash X^\rho, \Gamma)$, where X identifies $\nu X.\psi$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} X$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. Since any μ-variable lower than X is *not* active in X, we have $\mathcal{M}, s \models_{\sigma_{\rho \restriction X}} X$. This implies that $\mathcal{M}, s \models_{\sigma_{\rho \restriction X}} \psi$. Hence, σ is a good name signature for $\{\psi^{\rho \restriction X}\} \cup \Gamma$, which implies $\sigma \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^{\rho \restriction X}, \Gamma)$.

□

Lemma 4.28. *Suppose $\Theta \vdash Z^\rho, \Gamma$ is a goal such that Γ does* not *contain a formula $\gamma^{\rho'}$ in which Z is active and $\rho(Z) = \rho'(Z)$.*

$$\mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma) \succ_{\Theta''} \mathrm{Sig}(\Theta' \vdash \psi^{\rho \restriction Z}, \Gamma),$$

where Z identifies $\mu Z.\psi$, Θ' is Θ with all the names not *occurring in the latter goal removed, and Θ'' is any prefix of Θ' containing name $\rho(Z)$.*

Proof. Let σ be $\mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma)$. Hence, there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} Z$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'}$ in Γ. Since any μ-variable Z' lower than Z is *not* active in Z, we have $\mathcal{M}, s \models_{\sigma_{\rho \restriction Z}} Z$. Suppose $\rho(Z) = z$ and $\sigma_\rho(Z) = \sigma(z) = \alpha$. Thus we have

$$\mathcal{M}, s \models_{\sigma_{\rho \restriction Z}} \mu^\alpha Z.\psi \text{ and } \mathcal{M}, s \models_{\sigma'_{\rho \restriction Z}} \psi,$$

where $\sigma' = \sigma[z := \alpha']$ for some $\alpha' < \alpha$. Since, for each $\gamma^{\rho'} \in \Gamma$ in which Z is active, z does not occur in ρ', it follows that $\mathcal{M}, s \models_{\sigma'_{\rho'}} \gamma$, for each $\gamma^{\rho'} \in \Gamma$. Therefore, σ' is a good name signature for $\{\psi^{\rho[Z}\} \cup \Gamma$, and hence

$$\sigma' \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^{\rho[Z}, \Gamma).$$

For any prefix Θ'' of Θ' which contains z, we have $\sigma \succ_{\Theta''} \sigma'$ (because $\sigma(z) = \alpha > \sigma'(z)$), and therefore

$$\sigma \succ_{\Theta''} \mathrm{Sig}(\Theta' \vdash \psi^{\rho[Z}, \Gamma).$$

□

Theorem 4.29 (Completeness of ACON)**.** *Every satisfiable aconjunctive formula has a successful ACON-tableau.*

Proof. Suppose ϕ is a satisfiable, closed, aconjunctive formula. The construction of a successful ACON-tableau for ϕ starts with the smallest tableau $\mathcal{T}_0$ with only the initial goal $\vdash \phi$. We subsequently expand $\mathcal{T}_0$ while making sure the set of the formulae in each goal is satisfiable (the initial formula ϕ is satisfiable by assumption).

Suppose we have so far constructed $\mathcal{T}_0, ..., \mathcal{T}_i$. For each non-terminal leaf $u : \Theta \vdash \Gamma$ in $\mathcal{T}_i$, one or more of the following cases applies. Pick one applicable case and perform the described action, given priority to the earlier cases.

- $\Gamma = \psi^{\rho_1}, \psi^{\rho_2}, \Gamma'$. Apply Thin to create the subgoal $\Theta' \vdash \psi^{\rho_i}, \Gamma'$, for some $i \in \{1, 2\}$. By Lemma 4.27(a),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^{\rho_i}, \Gamma').$$

- $\Gamma = (\psi_1 \wedge \psi_2)^\rho, \Gamma'$. Apply R$\wedge$ to create the subgoal $\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma'$. By Lemma 4.27(b),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta} \mathrm{Sig}(\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma').$$

- $\Gamma = (\psi_1 \vee \psi_2)^\rho, \Gamma'$. Rule R$\vee$ can be applied to create either $\Theta \vdash \psi_1^\rho, \Gamma'$ or $\Theta \vdash \psi_2^\rho, \Gamma'$. By Lemma 4.27(c), there is a *least* i such that ψ_i, Γ' is satisfiable and

$$\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta} \mathrm{Sig}(\Theta \vdash \psi_i^\rho, \Gamma').$$

  Apply R$\vee$ to create the i-th subgoal.
- $\Gamma = \mu Z.\psi^\rho, \Gamma'$. Apply R$\mu$ to create the subgoal $\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma'$ where z^i is the first name for Z not occurring in Θ. By Lemma 4.27(d),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta} \mathrm{Sig}(\Theta \cdot z^i \vdash Z^{\rho \cdot z^i}, \Gamma').$$

- $\Gamma = (\nu X.\psi)^\rho, \Gamma'$. Apply $\mathsf{R}\nu$ to create the subgoal $\Theta \vdash X^\rho, \Gamma'$. By Lemma 4.27(e),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_\Theta \quad \mathrm{Sig}(\Theta \vdash X^\rho, \Gamma').$$

- $\Gamma = Z^\rho, \Gamma'$, where Z identifies $\mu Z.\psi$. Apply Unfold_μ to create subgoal $\Theta' \vdash \psi^{\rho\restriction Z}, \Gamma'$. By Lemma 4.9 and 4.28, for any prefix Θ'' of Θ' which contains $\rho(Z)$

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succ_{\Theta''} \quad \mathrm{Sig}(\Theta' \vdash \psi^{\rho\restriction Z}\}, \Gamma').$$

- $\Gamma = X^\rho, \Gamma'$, where X identifies $\nu X.\psi$. Apply Unfold_ν to create the subgoal $\Theta' \vdash \psi^{\rho\restriction X}, \Gamma'$. By Lemma 4.27(f),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta'} \quad \mathrm{Sig}(\Theta' \vdash \psi^{\rho\restriction X}, \Gamma').$$

- $\Gamma = (\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma'$ where $n \geq 1$ and Γ' contains only literals and/or $[\cdot]$-formulae. Apply $\mathsf{R}\langle\rangle$ to create n subgoals $\Theta_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}$ $(1 \leq i \leq n)$. By Lemma 4.27(g), each of these subgoals is satisfiable and

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta_i} \quad \mathrm{Sig}(\Theta_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}).$$

The construction must terminate at some tableau $\mathcal{T}'$ all whose leaves are terminal. Clearly since each goal in $\mathcal{T}'$ is satisfiable, all the leaves which contain only literals and/or $[\cdot]$-formulae are successful. Other leaves in $\mathcal{T}'$ must also be successful. Assume otherwise. Suppose $u_1 : \Theta_1 \vdash \Gamma_1, ..., u_n : \Theta_n \vdash \Gamma_n$ is the path to an unsuccessful leaf u_n from its companion u_1 (hence $\Theta_1 = \Theta_n$ and $\Gamma_1 = \Gamma_n$). Thus

- a μ-variable Z^ρ is unfolded at some node u_j, and
- the name $\rho(Z)$ occurs in each Γ_i.

Suppose $\Theta = z_1...z_k$, where $z_k = \rho(Z)$, is the prefix of Θ_1 up to the occurrence of $\rho(Z)$. Since $\Theta_1 = \Theta_n$, each z_i must also occur throughout the path, for if z_i is removed at some point, z_i cannot occur before z_k in Θ_n. For the same reason, no name other than $z_1, ..., z_{k-1}$ may occur before z_k in each Θ_i on the path. This means that Θ is a prefix of each Θ_i. It follows from the above construction that

$$\mathrm{Sig}(\Theta_1 \vdash \Gamma_1) \succeq_\Theta ... \succeq_\Theta \mathrm{Sig}(\Theta_n \vdash \Gamma_n).$$

Since rule Unfold_μ is applied to Z^ρ at u_j, by Lemma 4.27(f),

$$\mathrm{Sig}(\Theta_j \vdash \Gamma_j) \quad \succ_\Theta \quad \mathrm{Sig}(\Theta_{j+1} \vdash \Gamma_{j+1}).$$

This is impossible because $\Theta_1 = \Theta_n$ and $\Gamma_1 = \Gamma_n$. Therefore u_n must be successful. Hence every terminal in $\mathcal{T}'$ is successful. $\mathcal{T}'$ is thus a successful ACON-tableau for ϕ. □

4.2.3 Relation to Kozen's Tableau System

ACON is essentially a reformulation of Kozen's tableau system [Koz83]. The structure of a goal in Kozen's system is very similar to the structure of a goal in ACON. In particular, the label of each node in Kozen's tableaux consists of three components: a global sequence G of *counters*, a set Γ of formulae, and a function C which assigns a counter in G to each pair (ψ, X), where $\psi \in \Gamma$ and X is a variable active in ψ. Counters have two roles in Kozen's tableaux. One role is similar to the use of names in ACON, i.e. a counter is used to identify an occurrence of a μ-formula as it appears in the tableau. Secondly, each counter is used to count the number of the unfoldings of the μ-formula identified by the counter. The tableau system unsuccessfully terminates when any counter exceeds $2^{|\phi|}$. This counting-based termination condition is clearly inefficient in practice. In order to reject a tableau, one *always* has to wait until one of the counters exceeds the stated bound (by contrast, a tableau in ACON may be declared unsuccessful without having to unfold a μ-variable as many times). ACON avoids counting by reusing names. This allows us to bound the number of possible goals. A branch in an ACON-tableau can terminate as soon as there are two nodes labelled with the same goal on the branch. Although the worst-case complexity of ACON is no better than Kozen's tableau system, we find that, in practice, a tableau in ACON usually terminates earlier than a similar tableau in Kozen's system (where the same R$\vee$-choices are made in both tableaux).

Apart from this practical advantage (and, perhaps, a clearer presentation over Kozen's original formulation), the tableau system ACON offers us an insight in designing a tableau system which is sound and complete for the full logic. We describe such a tableau system in the next section.

4.3 Tableau System TS

We now present a tableau system for satisfiability which is sound and complete for all (guarded) formulae. In essence, the following tableau system TS generalises the idea in ACON, i.e. using names to record trail history. Instead of associating (at most) one name for each μ-variable to each formula in the tableau, a sequence of names for such μ-variable is used. In particular, a name is introduced when rule Unfold_μ is applied. The challenging problems are how to manage such sequences of names so that the number of possible goals is bounded and how to define termination and success conditions using the recorded name sequences. To solve these, we employ the idea from the determinisation construction for ω-word automata by Safra [Saf88]. In particular, a new tableau rule called Reset is introduced. The roles of Reset rule are twofold. First, systematic applications of Reset ensure that the sequences of names associated to the formulae in the tableau never become too long. Secondly, the success of a terminal can be determined by the existence of an application of the Reset rule on the branch. We

now describe the tableau system TS in detail.

Let ϕ be a closed and guarded formula. As before, we assume a linear ordering $X_1, ..., X_n$ of all the variables in ϕ such that X_i higher than X_j implies $i < j$. For each μ-variable Z in ϕ, we assume a sequence $z^1, z^2, ...$ of *names for* Z. The number of names required to build a tableau for ϕ in TS is bounded by the length of ϕ.

A *goal* in a TS-tableau for ϕ is a sequent of the form $\Theta \vdash \Gamma$ where

- Θ is a sequence of distinct names, called a *global sequence*, and
- Γ is a set of *augmented formulae* of the form ψ^ρ where ψ is a subformula of ϕ and ρ is a sequence of distinct names all of which occur in Θ. As before, we require that each name in Θ must occur in some sequence ρ in Γ.

The *initial goal* is $\vdash \phi$ (i.e. the global sequence is empty and so is the sequence of names augmenting ϕ).

Ordering of names. The ordering $<_\Theta$ on names and the ordering $\prec_\Theta$ on sequences of names with respect to a global sequence Θ are as defined in Definition 4.3 in the previous section. The restriction operations $\rho \restriction X$ and $\rho \restriction n$ are as given earlier.

Tableau rules. The tableau rules are given below. In the subgoals for rules Unfold_μ, Unfold_ν, Thin, and Reset_z, Θ' denotes the result of removing all the names in Θ not occurring in any augmented formula in the subgoal; similarly for Θ_i in the i-th subgoal of the rule $\mathsf{R}\langle\rangle$. See remarks below for some further explanation of the tableau rules.

$$\mathsf{R}\wedge: \quad \frac{\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma}{\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma}$$

$$\mathsf{R}\vee: \quad \frac{\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma}{\Theta \vdash \psi_i^\rho, \Gamma} \quad i \in \{1, 2\}$$

$$\mathsf{R}\sigma: \quad \frac{\Theta \vdash (\sigma X.\psi)^\rho, \Gamma}{\Theta \vdash X^\rho, \Gamma} \quad \sigma \in \{\mu, \nu\}$$

$$\mathsf{Unfold}_\mu: \quad \frac{\Theta \vdash Z^\rho, \Gamma}{\Theta' \cdot z^i \vdash \psi^{(\rho \restriction Z) \cdot z^i}, \Gamma} \quad \begin{array}{l} \text{if } Z \text{ identifies } \mu Z.\psi \text{ and} \\ z^i \text{ is the } \textit{first} \text{ name for } Z \textit{ not} \text{ occurring in } \Theta. \end{array}$$

$$\mathsf{Unfold}_\nu: \quad \frac{\Theta \vdash X^\rho, \Gamma}{\Theta' \vdash \psi^{\rho \restriction X}, \Gamma} \quad X \text{ identifies } \nu X.\psi.$$

$$\mathsf{R}\langle\rangle: \quad \frac{\Theta \vdash (\langle a_1\rangle\psi_1)^{\rho_1}, ..., (\langle a_n\rangle\psi_n)^{\rho_n}, \Gamma}{\Theta_1 \vdash \psi_1^{\rho_1}, \Gamma_{a_1} \mid ... \mid \Theta_n \vdash \psi_n^{\rho_n}, \Gamma_{a_n}} \quad n \geq 1$$

where

- Γ contains only literals and $[\cdot]$-formulae, and
- for each action a, $\Gamma_a = \{\psi^\rho \mid ([a]\psi)^\rho \in \Gamma\}$.

$$\mathsf{Thin}: \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma}$$

if $\rho \prec_\Theta \rho'$ or,
for some μ-variable Z, $\rho' \restriction Z$ is a *proper* prefix of $\rho \restriction Z$.

$$\mathsf{Reset}_z: \quad \frac{\Theta \vdash \psi_1^{\rho\cdot z\cdot z_1\cdot\rho_1}, ..., \psi_n^{\rho\cdot z\cdot z_n\cdot\rho_n}, \Gamma}{\Theta' \vdash \psi_1^{\rho\cdot z}, ..., \psi_n^{\rho\cdot z}, \Gamma} \quad n \geq 1$$

where $z, z_1, ..., z_n$ are names for the same variable and z does *not* occur in Γ.

Remark 4.30.

(1) In rule Unfold_μ, a new name for μ-variable Z is added to the global sequence. In order to bound the number of possible goals, we always choose the first name z^i for such μ-variable (i.e. one with the least i) not occurring in Θ.

(2) As before, the role of the *thinning rule* Thin is to eliminate redundant formulae in the goal. As shown below, for any formulae $\psi^\rho, \psi^{\rho'}$ (where $\rho \neq \rho'$) in a goal, the condition specified in the rule Thin uniquely chooses one of these formulae to discard.

Lemma 4.31. *Suppose Θ, ρ, ρ' are any sequences of names such that $\rho \neq \rho'$ and each name appearing in ρ or ρ' also appears in Θ. Then exactly one of the following holds:*

(a) $\rho \restriction Z$ is a proper prefix of $\rho' \restriction Z$, for some μ-variable Z.
(b) $\rho' \restriction Z$ is a proper prefix of $\rho \restriction Z$, for some μ-variable Z.
(c) $\rho \prec_\Theta \rho'$.
(d) $\rho' \prec_\Theta \rho$.

Proof. Since $\rho \neq \rho'$, either one of these is a proper prefix of the other or there must be some j such that $\rho(j) \neq \rho'(j)$ and $\rho(i) = \rho'(i)$ for each $i < j$. In the former case, if ρ is a proper prefix of ρ', then (a) obviously holds; otherwise (b) holds. In the latter case, if $\rho(j)$ and $\rho(j')$ are names for the *same* variable and $\rho(j) <_\Theta \rho'(j)$, then, by definition, (c) holds; similarly if $\rho'(j) <_\Theta \rho(j)$ then (d) holds. On the other hand, suppose $\rho(j)$ and $\rho'(j)$ are names for *different* variables, say Z and Z' respectively. If Z appears earlier than Z' in the assumed sequence $X_1, ..., X_n$, then $\rho' \restriction Z$ is a proper prefix of $\rho \restriction Z$ in which case (b) holds, otherwise $\rho \restriction Z$ is a proper prefix of $\rho' \restriction Z$ and thus (a) holds.

It is obvious that (a) and (b) cannot be true at the same time; similarly, for (c) and (d). If (a) or (b) is true, then there must be some j such that $\rho(j)$ and $\rho'(j)$ are names for *different* variables and $\rho(i) = \rho'(i)$ for each $i < j$. Hence, ρ and ρ' are $\prec_\Theta$-incomparable. If ρ and ρ' are $\prec_\Theta$-comparable, then, by definition, there is some j such that $\rho(j) \neq \rho'(j)$ are names for the *same* variable and $\rho(i) = \rho'(i)$ for each $i < j$. Clearly, this means that neither (a) nor (b) holds. Therefore, exactly one of the four cases above is true. □

(3) The family of *reset rules* Reset_z, where z ranges over names, ensures that the sequences of names augmenting the formulae in the goal do not become arbitrarily long. It is also essential in determining whether a terminal is successful. When not specifying the name z, we simply write Reset to refer to a reset rule in general.

(4) It is quite clear from the tableau rules that the name sequences appearing in a goal will have a special form as described below.

Observation 4.32. *Suppose ψ^ρ is an augmented formula in a goal $\Theta \vdash \Gamma$:*

- *ρ can be decomposed into $\rho(Z_1) \cdot ... \cdot \rho(Z_n)$ where each $\rho(Z_i)$ is a sequence of names for Z_i and $Z_1, ..., Z_n$ ($n \geq 0$) are some μ-variables in ϕ such that Z_i is higher than Z_{i+1}, for each $i < n$.*
- *The ordering of names in ρ is compatible with that in Θ.*
- *For any name z and formulae $\psi_1^{\rho_1}, \psi_2^{\rho_2}$ in Γ, if both ρ_1 and ρ_2 contain z, then the prefixes[2] of ρ_1 and ρ_2 up to the occurrence of z are equal.*

The last property implies that each name z occurring in Γ *uniquely* identifies a sequence $\rho \cdot z$ such that every sequence ρ' in Γ which contains z must have $\rho \cdot z$ as a prefix.

Restriction. To guarantee finiteness, when constructing a tableau it is required that rule Thin has the highest priority, followed by rule Reset, i.e. rule Thin is always applied whenever possible, and in case Thin is not applicable, rule Reset_z (for any name z) is applied if possible.

Termination. A *terminal* is a leaf $u : \Theta \vdash \Gamma$ such that *one* of the following conditions hold:

T1. Γ contains a complementary pair of literals.
T2. Γ is a consistent set of literals and $[\cdot]$-formulae.
T3. u has a proper ancestor $v : \Theta \vdash \Gamma$, called the *companion* of u.

When a terminal is reached, we stop applying a tableau rule to that node.

[2] As a convention throughout the thesis, every sequence is considered a prefix of itself.

Success. An *unsuccessful terminal* is a terminal $u : \Theta \vdash \Gamma$ such that *one* of the following holds.

U1. u satisfies T1 (i.e. Γ contains a complementary pair of literals).

U2. u has a companion v such that, for some name z, rule Reset_z is applied between v and u, and z occurs in each goal on the path from v to u.

A terminal is said to be *successful* otherwise. In other words, a *successful terminal* is a terminal $u : \Theta \vdash \Gamma$ such that *one* of the following holds.

S1. Γ is a consistent set of literals and $[\cdot]$-formulae.

S2. u has a companion v such that, for each name z, if rule Reset_z is applied between v and u, then there is a goal on the path from v to u where z does *not* occur.

A *successful tableau* $\mathcal{T}$ is a *finite* tableau all whose leaves are successful terminals. We consider some examples of TS-tableaux.

Example 4.33. The formula $\mu Z.\nu X.\langle a\rangle Z \wedge [a]X$ is clearly unsatisfiable. As expected, every tableau for this formula is unsuccessful. One such tableau is shown in Figure 4.6. Node 11 is a terminal with node 4 as the companion. It is an unsuccessful terminal because the name z^1 occurs in every goal from node 4 to node 11 and the rule Reset_{z^1} is applied at node 10. Observe that the non-success of the tableau reflects the fact that there is a μ-trail which goes through node 4 and node 11:

$$(4, X^{z^1}) \to (5, (\langle a\rangle Z \wedge [a]X)^{z^1}) \to (6, \langle a\rangle Z^{z^1}) \to (7, Z^{z^1}) \to$$
$$(8, (\nu X.\langle a\rangle Z \wedge [a]X)^{z^1 z^2}) \to (9, X^{z^1 z^2}) \to (10, X^{z^1 z^2}) \to (11, X^{z^1}) \to (4, X^{z^1}) \to ...$$

Another example is the unsatisfiable formula $\nu X.\mu Z.(\langle a\rangle Z \wedge [a]X)$. An unsuccessful tableau for this formula is shown in Figure 4.7. Node 13 is a terminal with node 5 as the companion. The name z^1 occurs in every goal from node 5 to node 13 and the rule Reset_{z^1} is applied at node 9. Hence, by condition U2, the terminal node 13 is unsuccessful.

Example 4.34. Consider the satisfiable formula

$$\neg P \wedge \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z)$$

in Example 4.6. A successful TS-tableau for this formula is shown in Figure 4.8. Node 14 is a successful terminal with node 9 as its companion.

Example 4.35. Consider the formula ϕ in Example 4.7, i.e. the conjunction of the formulae

$$\neg Q$$
$$\mu Z.(P \vee \langle a\rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)$$
$$\mu Y.(\neg P \vee \langle a\rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)$$

$$
\begin{array}{rc}
1: & \dfrac{\vdash \mu Z.\nu X.\langle a\rangle Z \wedge [a]X}{}\ \mathsf{R}\mu \\
2: & \dfrac{\vdash Z}{}\ \mathsf{Unfold}_{\mu} \\
3: & \dfrac{z^1 \vdash \nu X.\langle a\rangle Z \wedge [a]X^{z^1}}{}\ \mathsf{R}\nu \\
4: & \dfrac{z^1 \vdash X^{z^1}}{}\ \mathsf{Unfold}_{\nu} \\
5: & \dfrac{z^1 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1}}{}\ \mathsf{R}\wedge \\
6: & \dfrac{z^1 \vdash \langle a\rangle Z^{z^1},\ [a]X^{z^1}}{}\ \mathsf{R}\langle\rangle \\
7: & \dfrac{z^1 \vdash Z^{z^1},\ X^{z^1}}{}\ \mathsf{Unfold}_{\mu} \\
8: & \dfrac{z^1z^2 \vdash (\nu X.\langle a\rangle Z \wedge [a]X)^{z^1z^2},\ X^{z^1}}{}\ \mathsf{R}\nu \\
9: & \dfrac{z^1z^2 \vdash X^{z^1z^2},\ X^{z^1}}{}\ \mathsf{Thin} \\
10: & \dfrac{z^1z^2 \vdash X^{z^1z^2}}{}\ \mathsf{Reset}_{z^1} \\
11: & \dfrac{z^1 \vdash X^{z^1}}{} \\
 & \text{UNSUCCESSFUL}
\end{array}
$$

Figure 4.6: An unsuccessful TS-tableau for Example 4.33. Node 11 is a terminal with node 4 as the companion (condition T3). Since the name z^1 occurs in every goal from node 4 to node 11 and the rule Reset_{z^1} is applied at node 10, by condition U2, the terminal node 11 is unsuccessful.

$$
\begin{array}{rc}
1: & \dfrac{\vdash \nu X.\mu Z.(\langle a\rangle Z \wedge [a]X)}{}\ \mathsf{R}\nu \\
2: & \dfrac{\vdash X}{}\ \mathsf{Unfold}_{\nu} \\
3: & \dfrac{\vdash \mu Z.(\langle a\rangle Z \wedge [a]X)}{}\ \mathsf{R}\mu \\
4: & \dfrac{\vdash Z}{}\ \mathsf{Unfold}_{\mu} \\
5: & \dfrac{z^1 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1}}{}\ \mathsf{R}\wedge \\
6: & \dfrac{z^1 \vdash \langle a\rangle Z^{z^1},\ [a]X^{z^1}}{}\ \mathsf{R}\langle\rangle \\
7: & \dfrac{z^1 \vdash Z^{z^1},\ X^{z^1}}{}\ \mathsf{Unfold}_{\mu} \\
8: & \dfrac{z^1z^2 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1z^2},\ X^{z^1}}{}\ \mathsf{Unfold}_{\nu} \\
9: & \dfrac{z^1z^2 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1z^2},\ \mu Z.(\langle a\rangle Z \wedge [a]X)}{}\ \mathsf{Reset}_{z^1} \\
10: & \dfrac{z^1 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1},\ \mu Z.(\langle a\rangle Z \wedge [a]X)}{}\ \mathsf{R}\mu \\
11: & \dfrac{z^1 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1},\ Z}{}\ \mathsf{Unfold}_{\mu} \\
12: & \dfrac{z^1z^2 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1},\ (\langle a\rangle Z \wedge [a]X)^{z^2}}{}\ \mathsf{Thin} \\
13: & \dfrac{z^1 \vdash (\langle a\rangle Z \wedge [a]X)^{z^1}}{} \\
 & \text{UNSUCCESSFUL}
\end{array}
$$

Figure 4.7: Another unsuccessful TS-tableau for Example 4.33. Node 13 is a terminal with node 5 as the companion (condition T3). Since the name z^1 occurs in every goal from node 5 to node 13 and the rule Reset_{z^1} is applied at node 9, by condition U2, the terminal node 13 is unsuccessful.

$$
\begin{array}{ll}
1: & \dfrac{\vdash \neg P \wedge \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z)}{} \ \mathsf{R}\wedge \\
2: & \dfrac{\vdash \mu Z.P \vee \nu X.([a]X \wedge \langle a\rangle Z), \neg P}{} \ \mathsf{R}\mu \\
3: & \dfrac{\vdash Z, \neg P}{} \ \mathsf{Unfold}_{\mu} \\
4: & \dfrac{z^1 \vdash (P \vee \nu X.([a]X \wedge \langle a\rangle Z))^{z^1}, \neg P}{} \ \mathsf{R}\vee \\
5: & \dfrac{z^1 \vdash \nu X.([a]X \wedge \langle a\rangle Z)^{z^1}, \neg P}{} \ \mathsf{R}\nu \\
6: & \dfrac{z^1 \vdash X^{z^1}, \neg P}{} \ \mathsf{R}\nu \\
7: & \dfrac{z^1 \vdash ([a]X \wedge \langle a\rangle Z)^{z^1}, \neg P}{} \ \mathsf{R}\wedge \\
8: & \dfrac{\!\!\dfrac{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, \neg P}{}\!\!}{} \ \mathsf{R}\langle\rangle \\
9: & \dfrac{z^1 \vdash X^{z^1}, Z^{z^1}}{} \ \mathsf{Unfold}_{\nu} \\
10: & \dfrac{z^1 \vdash ([a]X \wedge \langle a\rangle Z)^{z^1}, Z^{z^1}}{} \ \mathsf{R}\wedge \\
11: & \dfrac{z^1 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, Z^{z^1}}{} \ \mathsf{Unfold}_{\mu} \\
12: & \dfrac{z^1 z^2 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, (P \vee \nu X.([a]X \wedge \langle a\rangle Z))^{z^1 z^2}}{} \ \mathsf{R}\vee \\
13: & \dfrac{\!\!\dfrac{z^1 z^2 \vdash [a]X^{z^1}, \langle a\rangle Z^{z^1}, P^{z^1 z^2}}{}\!\!}{} \ \mathsf{R}\langle\rangle \\
14: & \dfrac{z^1 \vdash X^{z^1}, Z^{z^1}}{\text{SUCCESSFUL}}
\end{array}
$$

Figure 4.8: A successful TS-tableau for Example 4.34. Node 14 is a terminal with node 9 as its companion (condition T3). Since the reset rule is not applied between these two nodes, by condition S2, node 14 is a successful terminal.

ϕ has a successful tableau in TS as shown in Figure 4.9. Node 32 is a successful terminal with node 12 as its companion. Recall that the ACON-tableau for this formula in Figure 4.5 prematurely terminates at node 19. This is no longer the case as the goals at nodes 11 and 19 are now distinct.

1: $\dfrac{\vdash \neg Q, \mu Z.(P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1), \mu Y.(\neg P \vee \langle a \rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)}{}\ \mathsf{R}\mu$

$\vdots$

10: $\dfrac{z^1y^1 \vdash \neg Q, P^{z^1}, [a]Z^{z^1}, [a]X_1^{z^1}, \langle a \rangle Y^{y^1}, [a]Y^{y^1}, [a]X_2^{y^1}}{}\ \mathsf{R}\langle\rangle$

11: $\dfrac{z^1y^1 \vdash Z^{z^1}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

12: $\dfrac{z^1y^1z^2 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

13: $\dfrac{z^1y^1z^2y^2 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, (\neg P \vee \langle a \rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{R}\wedge$

14: $\dfrac{z^1y^1z^2y^2 \vdash (P \vee \langle a \rangle Z)^{z^1z^2}, (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, (\neg P \vee \langle a \rangle Y)^{y^1y^2}, (Q \vee \nu X_2.[a]Y \wedge [a]X_2)^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{R}\vee$

15: $\dfrac{z^1y^1z^2y^2 \vdash P^{z^1z^2}, Q^{z^1z^2}, X_1^{z^1}, \langle a \rangle Y^{y^1y^2}, Q^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Thin}$

16: $\dfrac{z^1y^1z^2y^2 \vdash P^{z^1z^2}, Q^{z^1z^2}, X_1^{z^1}, \langle a \rangle Y^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Unfold}_\nu$

17: $\dfrac{z^1y^1z^2y^2 \vdash P^{z^1z^2}, Q^{z^1z^2}, ([a]Z \wedge [a]X_1)^{z^1}, \langle a \rangle Y^{y^1y^2}, ([a]Y \wedge [a]X_2)^{y^1}}{}\ \mathsf{R}\wedge$

18: $\dfrac{z^1y^1z^2y^2 \vdash P^{z^1z^2}, Q^{z^1z^2}, [a]Z^{z^1}, [a]X_1^{z^1}, \langle a \rangle Y^{y^1y^2}, [a]Y^{y^1}, [a]X_2^{y^1}}{}\ \mathsf{R}\langle\rangle$

19: $\dfrac{z^1y^1y^2 \vdash Z^{z^1}, X_1^{z^1}, Y^{y^1y^2}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Thin}$

20: $\dfrac{z^1y^1y^2 \vdash Z^{z^1}, X_1^{z^1}, Y^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

21: $\dfrac{z^1y^1y^2z^2 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, Y^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

22: $\dfrac{z^1y^1y^2z^2y^3 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, (\neg P \vee \langle a \rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)^{y^1y^2y^3}, X_2^{y^1}}{}\ \mathsf{Reset}_{y^2}$

23: $\dfrac{z^1y^1y^2z^2 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, (\neg P \vee \langle a \rangle Y) \wedge (Q \vee \nu X_2.[a]Y \wedge [a]X_2)^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{R}\wedge$

24: $\dfrac{z^1y^1y^2z^2 \vdash (P \vee \langle a \rangle Z)^{z^1z^2}, (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, (\neg P \vee \langle a \rangle Y)^{y^1y^2}, (Q \vee \nu X_2.[a]Y \wedge [a]X_2)^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{R}\vee$

25: $\dfrac{z^1y^1y^2z^2 \vdash \langle a \rangle Z^{z^1z^2}, Q^{z^1z^2}, X_1^{z^1}, \neg P^{y^1y^2}, Q^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Thin}$

26: $\dfrac{z^1y^1y^2z^2 \vdash \langle a \rangle Z^{z^1z^2}, Q^{z^1z^2}, X_1^{z^1}, \neg P^{y^1y^2}, X_2^{y^1}}{}\ \mathsf{Unfold}_\nu$

27: $\dfrac{z^1y^1y^2z^2 \vdash \langle a \rangle Z^{z^1z^2}, Q^{z^1z^2}, ([a]Z \wedge [a]X_1)^{z^1}, \neg P^{y^1y^2}, ([a]Y \wedge [a]X_2)^{y^1}}{}\ \mathsf{R}\wedge$

28: $\dfrac{z^1y^1y^2z^2 \vdash \langle a \rangle Z^{z^1z^2}, Q^{z^1z^2}, [a]Z^{z^1}, [a]X_1^{z^1}, \neg P^{y^1y^2}, [a]Y^{y^1}, [a]X_2^{y^1}}{}\ \mathsf{R}\langle\rangle$

29: $\dfrac{z^1y^1z^2 \vdash Z^{z^1z^2}, Z^{z^1}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Thin}$

30: $\dfrac{z^1y^1z^2 \vdash Z^{z^1z^2}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Unfold}_\mu$

31: $\dfrac{z^1y^1z^2z^3 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2z^3}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}\ \mathsf{Reset}_{z^2}$

32: $\dfrac{z^1y^1z^2 \vdash (P \vee \langle a \rangle Z) \wedge (Q \vee \nu X_1.[a]Z \wedge [a]X_1)^{z^1z^2}, X_1^{z^1}, Y^{y^1}, X_2^{y^1}}{}$

SUCCESSFUL

Figure 4.9: A successful TS-tableau for the formula ϕ in Example 4.35. Node 32 is a terminal with node 12 as its companion (condition T3). Although Reset_{y^2} and Reset_{z^2} are applied between node 12 and node 32, the names y^2 and z^2 are removed at some points between these two nodes. Hence, by condition S2, node 32 is a successful terminal.

Remark 4.36. Before we proceed with the soundness and completeness proofs, let us say a few words about our definition of the thinning rule. The condition in the thinning rule Thin (in particular, the ordering $\prec_\Theta$) may look strange at first. One may ask why a simpler ordering, such as the standard lexicographical ordering, is not used instead. We shall look at two simpler candidates for the thinning rule and show how they fail the soundness.

First, consider the standard lexicographical ordering $<_\Theta$ with respect to Θ, i.e. $\rho <_\Theta \rho'$ iff either ρ is a proper prefix of ρ' or there is some j such that $\rho(j) <_\Theta \rho'(j)$ and $\rho(i) = \rho'(i)$ for each $i < j$. A natural thinning rule based on $<_\Theta$ is the following:

$$\mathsf{Thin}_1 : \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma} \quad \text{if } \rho <_\Theta \rho'.$$

It is not difficult to show that the tableau system becomes unsound if Thin_1 is used in place of Thin. The following is a simple counterexample.

Example 4.37. Consider the unsatisfiable formula $\mu Z.\nu X.\langle a\rangle(X \wedge Z)$. Figure 4.10(a) shows a tableau for this formula where the rule Thin_1 is used in place of Thin. This tableau is successful because there is no node between the terminal node 10 and its companion node 4 where the reset rule is applied. Notice that, at node 9, by keeping the formula X^{z^1} instead of $X^{z^1 z^2}$, we obtain a repeat in the branch without having to apply the reset rule.

On the other hand, if the rule Thin is used for thinning, the formula will have no successful tableaux. One example is the tableau in Figure 4.10(b). At node 9, the rule Thin keeps the formula $X^{z^1 z^2}$, after which Reset_{z^1} can be applied. Hence, unlike the first tableau, this tableau is unsuccessful.

In the tableau $\mathcal{T}_1$ in Figure 4.10(a), there are trails from X^{z^1} in node 4 to both X^{z^1} and $X^{z^1 z^2}$ in node 9. But the trail to $X^{z^1 z^2}$ goes through an unfolding of Z (between node 7 and node 8) whereas the trail to X^{z^1} does not. This is why the name sequence in $X^{z^1 z^2}$ contains an additional name z^2. By discarding $X^{z^1 z^2}$ from node 9, it is as if we "forget" that there is such a trail in which Z is unfolded. Consequently, the tableau is incorrectly classified as successful. This suggests that if a goal contains formulae $\psi^\rho, \psi^{\rho'}$ where ρ is a proper prefix of ρ', we should keep the formula $\psi^{\rho'}$ instead. We can modify the thinning condition to cover this case. Precisely, define the ordering $\prec'_\Theta$ as follows: $\rho \prec'_\Theta \rho'$ iff, for some j, $\rho(j) <_\Theta \rho'(j)$ and, for each $i < j$, $\rho(i) = \rho'(i)$. This ordering is *not* total. But it is easy to see that two distinct sequences ρ and ρ' are incomparable under $\prec'_\Theta$ if and only if one of them is a prefix of the other. We then define a modified thinning rule as follows:

$$\mathsf{Thin}_2 : \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma} \quad \text{if } \rho \prec'_\Theta \rho' \text{ or } \rho' \text{ is a proper prefix of } \rho.$$

With this modification, the rule Thin_2 can be safely used in the previous example.

$$
\begin{array}{ll}
1: & \dfrac{\vdash \mu Z.\nu X.\langle a\rangle(X \wedge Z)}{}\ \mathsf{R}\mu \\
2: & \dfrac{\vdash Z}{}\ \mathsf{Unfold}_{\mu} \\
3: & \dfrac{z^1 \vdash \nu X.\langle a\rangle(X \wedge Z)^{z^1}}{}\ \mathsf{R}\nu \\
4: & \dfrac{z^1 \vdash X^{z^1}}{}\ \mathsf{Unfold}_{\nu} \\
5: & \dfrac{z^1 \vdash \langle a\rangle(X \wedge Z)^{z^1}}{}\ \mathsf{R}\langle\rangle \\
6: & \dfrac{z^1 \vdash (X \wedge Z)^{z^1}}{}\ \mathsf{R}\wedge \\
7: & \dfrac{z^1 \vdash X^{z^1},\, Z^{z^1}}{}\ \mathsf{Unfold}_{\mu} \\
8: & \dfrac{z^1 z^2 \vdash X^{z^1},\, \nu X.\langle a\rangle(X \wedge Z)^{z^1 z^2}}{}\ \mathsf{R}\nu \\
9: & \dfrac{z^1 z^2 \vdash X^{z^1},\, X^{z^1 z^2}}{}\ \mathsf{Thin}_1 \\
10: & \dfrac{z^1 \vdash X^{z^1}}{} \\
 & \text{SUCCESSFUL}
\end{array}
$$

(a) Tableau $\mathcal{T}_1$

$$
\begin{array}{ll}
1: & \dfrac{\vdash \mu Z.\nu X.\langle a\rangle(X \wedge Z)}{}\ \mathsf{R}\mu \\
2: & \dfrac{\vdash Z}{}\ \mathsf{Unfold}_{\mu} \\
3: & \dfrac{z^1 \vdash \nu X.\langle a\rangle(X \wedge Z)^{z^1}}{}\ \mathsf{R}\nu \\
4: & \dfrac{z^1 \vdash X^{z^1}}{}\ \mathsf{Unfold}_{\nu} \\
5: & \dfrac{z^1 \vdash \langle a\rangle(X \wedge Z)^{z^1}}{}\ \mathsf{R}\langle\rangle \\
6: & \dfrac{z^1 \vdash (X \wedge Z)^{z^1}}{}\ \mathsf{R}\wedge \\
7: & \dfrac{z^1 \vdash X^{z^1},\, Z^{z^1}}{}\ \mathsf{Unfold}_{\mu} \\
8: & \dfrac{z^1 z^2 \vdash X^{z^1},\, \nu X.\langle a\rangle(X \wedge Z)^{z^1 z^2}}{}\ \mathsf{R}\nu \\
9: & \dfrac{z^1 z^2 \vdash X^{z^1},\, X^{z^1 z^2}}{}\ \mathsf{Thin} \\
10: & \dfrac{z^1 z^2 \vdash X^{z^1 z^2}}{}\ \mathsf{Reset}_{z^1} \\
11: & \dfrac{z^1 \vdash X^{z^1}}{} \\
 & \text{UNSUCCESSFUL}
\end{array}
$$

(b) Tableau $\mathcal{T}_2$

Figure 4.10: In tableau $\mathcal{T}_1$, the rule Thin_1 is used for thinning at node 9. The tableau is successful because there is no node between the terminal node 10 and its companion (node 4) where the reset rule is applied. On the other hand, if Thin is used, the formula $X^{z^1 z^2}$ will be chosen instead of X^{z^1}. The rule Reset_{z_1} can then be applied afterwards. This is shown in tableau $\mathcal{T}_2$. This tableau is hence unsuccessful because Reset_{z^1} is applied between the terminal node 11 and its companion (node 4) and z^1 occurs in every node between these two nodes.

However, Thin_2 may not always work in general and, in particular, when the initial formula has more than one variable. This is illustrated in the following example.

Example 4.38. Consider the *unsatisfiable* formula $\mu Z.\mu Y.\nu X.\langle a\rangle(Y \vee (X \wedge Z))$. In the tableau in Figure 4.11, Thin_2 is used for thinning at node 14. Since $z^1y^1 \prec_\Theta z^1z^2y^2$ (where $\Theta = z^1y^1z^2y^2$), the formula $X^{z^1z^2y^2}$ is discarded, after which the tableau terminates successfully. But if we compare the two sequences with respect to Z, $(z^1y^1) \restriction Z$ is a proper prefix of $(z^1z^2y^2) \restriction Z$. This reflects the fact that Z is unfolded on the trail from node 6 to $X^{z^1z^2y^2}$ in node 14, but not on the trail from node 6 to $X^{z^1y^1}$ in node 14. This suggests that $X^{z^1z^2y^2}$ should be kept instead.

$$
\begin{array}{ll}
1: & \dfrac{\vdash \mu Z.\mu Y.\nu X.\langle a\rangle(Y \vee (X \wedge Z))}{}\ \mathsf{R}\mu \\
2: & \dfrac{\vdash Z}{}\ \mathsf{Unfold}_\mu \\
3: & \dfrac{z^1 \vdash \mu Y.\nu X.\langle a\rangle(Y \vee (X \wedge Z))^{z^1}}{}\ \mathsf{R}\mu \\
4: & \dfrac{z^1 \vdash Y^{z^1}}{}\ \mathsf{Unfold}_\mu \\
5: & \dfrac{z^1y^1 \vdash \nu X.\langle a\rangle(Y \vee (X \wedge Z))^{z^1y^1}}{}\ \mathsf{R}\nu \\
6: & \dfrac{z^1y^1 \vdash X^{z^1y^1}}{}\ \mathsf{Unfold}_\nu \\
7: & \dfrac{z^1y^1 \vdash \langle a\rangle(Y \vee (X \wedge Z))^{z^1y^1}}{}\ \mathsf{R}\langle\rangle \\
8: & \dfrac{z^1y^1 \vdash (Y \vee (X \wedge Z))^{z^1y^1}}{}\ \mathsf{R}\vee \\
9: & \dfrac{z^1y^1 \vdash (X \wedge Z)^{z^1y^1}}{}\ \mathsf{R}\wedge \\
10: & \dfrac{z^1y^1 \vdash X^{z^1y^1}, Z^{z^1y^1}}{}\ \mathsf{Unfold}_\mu \\
11: & \dfrac{z^1y^1z^2 \vdash X^{z^1y^1}, \mu Y.\nu X.\langle a\rangle(Y \vee (X \wedge Z))^{z^1z^2}}{}\ \mathsf{R}\mu \\
12: & \dfrac{z^1y^1z^2 \vdash X^{z^1y^1}, Y^{z^1z^2}}{}\ \mathsf{Unfold}_\mu \\
13: & \dfrac{z^1y^1z^2y^2 \vdash X^{z^1y^1}, \nu X.\langle a\rangle(Y \vee (X \wedge Z))^{z^1z^2y^2}}{}\ \mathsf{R}\nu \\
14: & \dfrac{z^1y^1z^2y^2 \vdash X^{z^1y^1}, X^{z^1z^2y^2}}{}\ \mathsf{Thin}_2 \\
15: & \dfrac{z^1y^1 \vdash X^{z^1y^1}}{} \\
 & \text{SUCCESSFUL}
\end{array}
$$

Figure 4.11: In this tableau, the rule Thin_2 is applied at node 14. The tableau is successful because there is no node between the terminal node 15 and its companion (node 6) where the reset rule is applied. If Thin is used instead of Thin_2, the formula $X^{z^1z^2y^2}$ will be chosen instead of $X^{z^1y^1}$, and the resulting tableau will not terminate at node 15.

A correct way to decide which formula in a pair $\psi^\rho, \psi^{\rho'}$ to discard is by comparing the restrictions of ρ, ρ' with respect to each μ-variable in turn from $Z_1, ..., Z_m$ (where $Z_1, ..., Z_m$ is the sequence of the μ-variables in ϕ in the same order as the assumed linear ordering $X_1, ..., X_n$ of the variables in ϕ). That is, we start by comparing $\rho \restriction Z_1$ and $\rho' \restriction Z_1$ using $\prec'_\Theta$ as above. If $\rho \restriction Z_1 \prec'_\Theta \rho' \restriction Z_1$, we keep the formula ψ^ρ; and conversely

if $\rho' \restriction Z_1 \prec'_\Theta \rho \restriction Z_1$. If $\rho \restriction Z_1$ and $\rho' \restriction Z_1$ are not equal but incomparable under $\prec'_\Theta$, hence one of the sequences is a proper prefix of the other, we keep ψ^ρ (resp., $\psi^{\rho'}$) if $\rho \restriction Z_1$ (resp., $\rho' \restriction Z_1$) is the longer of the two. In case $\rho \restriction Z_1 = \rho' \restriction Z_1$, we proceed with Z_2 and compare $\rho \restriction Z_2$ and $\rho' \restriction Z_2$ in the same way, and so on. Since $\rho \neq \rho'$, we must eventually terminate at some Z_j. Thus, this is equivalent to the following thinning rule:

$$\mathsf{Thin}_3: \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma} \qquad \begin{array}{l} \text{if, for some } j,\ \rho \restriction Z_i = \rho' \restriction Z_i \text{ for each } i < j, \text{ and} \\ \text{either } \rho \restriction Z_j \prec'_\Theta \rho' \restriction Z_j \text{ or } \rho' \restriction Z_j \text{ is a proper prefix of } \rho \restriction Z_j. \end{array}$$

This rule is equivalent to Thin. To see this, observe that $\rho \prec_\Theta \rho'$ (where $\prec_\Theta$ is as described in Definition 4.3) iff, for some j, $\rho \restriction Z_j \prec'_\Theta \rho' \restriction Z_j$ and $\rho \restriction Z_i = \rho' \restriction Z_i$ for each $i < j$. Hence, it is clear that the thinning condition in Thin_3 is equivalent to that in Thin. We shall prove formally in the subsequent sections that the tableau system TS which uses Thin for thinning is both sound and complete.

Finiteness. It can be shown that every TS-tableau is *finite*. This follows from the restriction that rules Thin and Reset are applied whenever possible and from the canonical choice of a new name introduced by rule Unfold_μ. These two conditions ensure that there are finitely many possible goals in a tableau.

Lemma 4.39. *For each μ-variable Z, the names for Z occurring in each goal (in any tableau) are among $z^1, ..., z^{|\phi|}$.*

Proof. We can show that this property is an invariant when constructing a tableau for ϕ, provided that rules Thin and Reset are applied whenever possible. Suppose $\Theta \vdash \Gamma$ is a goal such that, for each μ-variable Z, the names for Z in the goal are among $z^1, ..., z^{|\phi|}$. Since every rule except Unfold_μ does not introduce a new name, this property still holds when such rule is applied. So suppose rule Unfold_μ is applied to a μ-variable Z in $\Theta \vdash \Gamma$; hence by our restriction, both rules Thin and Reset are not applicable. Suppose $\psi_1^{\rho_1}, ...\psi_n^{\rho_n}$ are all the formulae in Γ such that each ρ_i contains one or more names for Z. Since Thin is not applicable, $\psi_1, ..., \psi_n$ must be distinct. Hence, obviously, $n < |\phi|$ (because there are less than $|\phi|$ subformulae of ϕ in which Z is active). We claim that there must be some name z^i for Z ($1 \leq i \leq |\phi|$) not occurring in $\rho_1, ..., \rho_n$. If this is *not* the case, there must be some name z^j for Z which occurs before a name for Z in each ρ_i (i.e. for each ρ_i, z^j does *not* occur last among the names for Z in ρ_i). Hence the formulae in which z^j occurs are of the form $\varphi_1^{\rho \cdot z^j \cdot \rho'_1}, ..., \varphi_m^{\rho \cdot z^j \cdot \rho'_m}$ where each ρ'_i contains a name for Z. This implies that rule Reset_{z^j} is applicable, contradicting our assumption. Hence there must be a name for Z among $z^1, ..., z^{|\phi|}$ not occurring in the goal $\Theta \vdash \Gamma$. We may then choose the first such name when unfolding Z. □

Lemma 4.40. *There are $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$ possible goals in TS-tableaux for ϕ.*

Proof. For brevity, we let n denote the length of ϕ and m denote the number of μ-variables in ϕ. By a *goal*, we mean a goal in any TS-tableau for ϕ.

We begin by showing that the number of possible goals where the rule Thin is *not* applicable is bounded. To do so, we shall first count the number of distinct sets Γ such that $\Theta \vdash \Gamma$, for some Θ, is a such a goal. Let $\mathcal{C}$ be the collection of all such sets Γ. Observe that each set Γ in $\mathcal{C}$ has the following properties:

(1) For each $\psi \in \mathrm{Sub}(\phi)$, there is *at most* one formula of the form ψ^ρ in Γ.
(2) For any μ-variable Z, the names for Z appearing in Γ are among the first n ones, i.e. $z^1, ..., z^n$.
(3) For any name z appearing in Γ, there is a *unique* sequence $\rho \cdot z$, such that for any $\psi^{\rho'} \in \Gamma$, if z occurs in ρ', then $\rho \cdot z$ is a prefix of ρ'.

(1) follows from the fact that Γ belongs to a goal in which Thin is not applicable. (2) follows from the previous lemma. (3) is a trivial property of any goal (see Observation 4.32).

We shall represent each $\Gamma \in \mathcal{C}$ as a pair $(f_\Gamma, g_\Gamma) \in \mathcal{F}_\Gamma \times \mathcal{G}_\Gamma$, where

- $\mathcal{F}_\Gamma$ is the set of all *partial functions* from $\mathrm{Sub}(\phi)$ to $\mathsf{Name}_n \cup \{nil\}$, where Name_n contains the first n names of each μ-variable in ϕ and *nil* is some symbol distinct from every name;
- $\mathcal{G}_\Gamma$ is the set of all *partial functions* from Name_n to $\mathsf{Name}_n \cup \{nil\}$.

Namely, for each $\Gamma \in \mathcal{C}$, we define f_Γ and g_Γ as follows:

- $f_\Gamma(\psi) = nil$ if $\psi^\epsilon \in \Gamma$, where ϵ is the empty sequence;
- $f_\Gamma(\psi) = x$ if $\psi^{\rho \cdot x} \in \Gamma$, for some ρ;
- $f_\Gamma(\psi)$ is undefined if there is no ψ^ρ, for any ρ, in Γ;
- $g_\Gamma(z) = nil$ if, for some ψ, $\psi^z \in \Gamma$;
- $g_\Gamma(z) = y$ if, for some formula ψ and sequence ρ, $\psi^{\rho \cdot y \cdot z} \in \Gamma$;
- $g_\Gamma(z)$ is undefined if z does not appear in Γ.

Properties (1) - (3) above ensure that these definitions for f_Γ and g_Γ are well defined. It is easy to check that this representation is a one-one mapping from $\mathcal{C}$ into $\mathcal{F}_\Gamma \times \mathcal{G}_\Gamma$. To see this, suppose there are some sets $\Gamma_1 \neq \Gamma_2$ in $\mathcal{C}$ such that $f_{\Gamma_1} = f_{\Gamma_2}$ and $g_{\Gamma_1} = g_{\Gamma_2}$. Since the domains of f_{Γ_1} and f_{Γ_2} are the same, Γ_1 contains a formula ψ^ρ, for some ρ, iff Γ_2 also contains a formula $\psi^{\rho'}$, for some ρ'. And since $\Gamma_1 \neq \Gamma_2$, this means that there is some $\psi \in \mathrm{Sub}(\phi)$ such that $\psi^{\rho_1} \in \Gamma_1$ and $\psi^{\rho_2} \in \Gamma_2$ for some sequences $\rho_1 \neq \rho_2$. It is clear that ρ_1 and ρ_2 must be non-empty and end with the same name, for otherwise $f_{\Gamma_1}(\psi) \neq f_{\Gamma_2}(\psi)$. Suppose $z \cdot \rho$ is the longest common suffix of ρ_1 and ρ_2. Thus, we can write $\rho_1 = \rho_1' \cdot z \cdot \rho$ and $\rho_2 = \rho_2' \cdot z \cdot \rho$, for some sequences ρ_1', ρ_2' such that either ρ_1' and ρ_2' end with different names or one of them is empty while the other is not. But this would implies that $g_{\Gamma_1}(z) \neq g_{\Gamma_2}(z)$, contradicting our assumption. Therefore, the mapping is one-one. And hence $|\mathcal{C}|$ is bounded by the size of $\mathcal{F}_\Gamma \times \mathcal{G}_\Gamma$, which is $\leq (mn+2)^n \cdot (mn+2)^{mn}$.

Next, we take into account the ordering of names. Since the number of names occurring in each set $\Gamma \in \mathcal{C}$ is $\leq mn$, the names in each set Γ can be linearly ordered in at most $(mn)!$ ways. Therefore, the number of possible goals in which the rule Thin is not applicable is no greater than $(mn)! \cdot (mn+2)^n \cdot (mn+2)^{mn} = 2^{O(mn\log(mn))}$, which equals to $2^{O(mn\log(n))}$ because $m < n$.

It is not difficult to see that the number of all possible goals is not much larger than this bound. To see this, suppose $\Theta \vdash \Gamma$ is a goal where Thin is *not* applicable. Depending on which tableau rule is applied on $\Theta \vdash \Gamma$, which formula in Γ is reduced by the rule, and how the formula is reduced, there can be many possible subgoals of $\Theta \vdash \Gamma$. But it is quite obvious that the number of possible subgoals is less than $2n$. Suppose $\Theta' \vdash \Gamma'$ is a subgoal of $\Theta \vdash \Gamma$. The rule Thin may be applicable on $\Theta' \vdash \Gamma'$. As explained in the proof of lemma 4.11, if Thin is applicable on $\Theta' \vdash \Gamma'$, then one of the following holds:

- Γ' contains a redundant triple $\psi^{\rho_1}, \psi^{\rho_2}, \psi^{\rho_3}$ (where ρ_1, ρ_2, ρ_3 are distinct). In this case, there are three ways to apply Thin on $\Theta' \vdash \Gamma'$.
- Γ' contains two redundant pairs $\psi^{\rho_1}, \psi^{\rho_2}$ and $\psi'^{\rho_3}, \psi'^{\rho_4}$ (where $\psi \neq \psi'$, $\rho_1 \neq \rho_2$, $\rho_3 \neq \rho_4$). In this case, there are two ways to apply Thin on $\Theta' \vdash \Gamma'$.
- Γ' contains just one redundant pair $\psi^{\rho}, \psi^{\rho'}$ (where $\rho \neq \rho'$). In this case, there is only one way to apply Thin on $\Theta' \vdash \Gamma'$.

This means that there are at most three ways to apply Thin on $\Theta' \vdash \Gamma'$, and if $\Theta'' \vdash \Gamma''$ is a result of applying Thin on $\Theta' \vdash \Gamma'$, then either Thin is not applicable on $\Theta'' \vdash \Gamma''$ or Thin can be applied on $\Theta'' \vdash \Gamma''$ once to obtain a goal where Thin is *not* applicable. Thus, we may conclude that each goal $\Theta \vdash \Gamma$ where Thin is *not* applicable can generate at most $4 \cdot 2n$ possible goals where Thin is applicable. Therefore, the total number of all possible goals is bounded by $2^{O(mn\log(n))} + 8n \cdot 2^{O(mn\log(n))} = 2^{O(mn\log(n))}$. □

Lemma 4.41. *Every tableau for ϕ is a finite tree of degree $O(|\phi|)$ and height $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$.*

Proof. The degree of a tableau cannot exceed the number of $\langle\cdot\rangle$-subformulae of ϕ, and hence is bounded by $O(|\phi|)$. By the previous lemma, a branch in a tableau cannot be longer than $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$. □

4.3.1 Soundness

Suppose $\mathcal{T}$ is a successful TS-tableau for a guarded and closed formula ϕ. Let $\mathcal{M}_{\mathcal{T}}$ be the model corresponding to $\mathcal{T}$ as given by Definition 4.14 in the previous section. We show that $\mathcal{M}_{\mathcal{T}}$ is a model for ϕ. As before, we prove this by showing that a successful TS-tableau may not contain a μ-trail. The notion of trails on TS-tableaux can be given as on ACON-tableaux:

Definition 4.42 (Trails)**.** The *dependency relation* $\rightarrow$ on a TS-tableau $\mathcal{T}$ is the smallest (binary) relation over pairs (u, ψ^ρ), where u is a node and ψ^ρ is in the goal at u, satisfying the following:

(a) For each node u where $\mathsf{R}\wedge$, $\mathsf{R}\vee$, $\mathsf{R}\mu$, $\mathsf{R}\nu$, or Unfold_σ is applied, if the formula ψ^ρ in u is reduced to $\psi'^{\rho'}$ in the child u', then $(u, \psi^\rho) \rightarrow (u', \psi'^{\rho'})$.

(b) For each node u where Thin is applied, if the formulae $\psi^\rho, \psi^{\rho'}$ are reduced to ψ^ρ in the child u' (i.e. $\psi^{\rho'}$ is discarded), then $(u, \psi^\rho) \rightarrow (u', \psi^\rho)$ and $(u, \psi^{\rho'}) \rightarrow (u', \psi^\rho)$.

(c) For each node u where Reset_z is applied, if the formula $\psi_i^{\rho \cdot z \cdot z_i \cdot \rho_i}$ in u is reduced to $\psi_i^{\rho \cdot z}$ in the child u', then $(u, \psi_i^{\rho \cdot z \cdot z_i \cdot \rho_i}) \rightarrow (u', \psi_i^{\rho \cdot z})$.

(d) In the above cases, if a formula γ^ρ in u is not affected by the tableau rule (hence γ^ρ is also in the child u'), then $(u, \gamma^\rho) \rightarrow (u', \gamma^\rho)$.

(e) For each node u where $\mathsf{R}\langle\rangle$ is applied and $(\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma$ are the formulae in u, if the formula $(\langle a_i \rangle \psi_i)^{\rho_i}$ in u is reduced to $\psi_i^{\rho_i}$ in a child u_i, then $(u, (\langle a_i \rangle \psi_i)^{\rho_i}) \rightarrow (u_i, \psi_i^{\rho_i})$ and, for each formula $([a_i]\psi)^\rho \in \Gamma$, $(u, ([a_i]\psi)^\rho) \rightarrow (u_i, \psi^\rho)$.

(f) Lastly, for any terminal u which has a companion v, $(u, \psi^\rho) \rightarrow (v, \psi^\rho)$ for each ψ^ρ in u.

A *trail* in tableau $\mathcal{T}$ is a path over its dependency relation.

The following lemma concerns a comparison of the name sequences in consecutive nodes. It follows mainly from the fact that no tableau rules swap the ordering of names in a global sequence and, when a new name is introduced to a goal (i.e. by the rule Unfold_μ), it is appended to the end of the global sequence.

Lemma 4.43. *Suppose* $u_1 : \Theta_1 \vdash \Gamma_1, ..., u_n : \Theta_n \vdash \Gamma_n$ *are nodes where* $u_1 \Rightarrow ... \Rightarrow u_n$*. For any name sequences* $\rho_1, ..., \rho_n$*, if* $\rho_i \succeq_{\Theta_i} \rho_{i+1}$ *for each* i $(1 \leq i < n)$ *and* $\rho_n = \rho_1$ *then* $\rho_1 = ... = \rho_n$*.*

Proof. Suppose $\rho_1 \succeq_{\Theta_1} \rho_2 \succeq_{\Theta_2} ... \succeq_{\Theta_{n-1}} \rho_n = \rho_1$. Assume that *not* all of $\rho_1, ...\rho_n$ are equal. Clearly there must be some position m such that $\rho_1(m) \geq_{\Theta_1} \rho_2(m) \geq_{\Theta_2} ... \geq_{\Theta_{n-1}} \rho_n(m)$ where one or more of these inequalities is strict (here $\rho_i(m)$ denotes the m-th name in the sequence ρ_i). Since no tableau rules swap the ordering of names in the global sequences and new names are appended to the end of the global sequences, this implies that each $\rho_i(m)$ must occur in Θ_1. Particularly, we have $\rho_1(m) \geq_{\Theta_1} \rho_2(m) \geq_{\Theta_1} ... \geq_{\Theta_1} \rho_n(m)$. Since $\rho_n = \rho_1$, all the names $\rho_1(m), ..., \rho_n(m)$ must be equal, contradicting what previously assumed. Hence, it must be the case that $\rho_1 = ... = \rho_n$. □

Lemma 4.44. *Every successful* TS*-tableau does* not *contain a* μ*-trail.*

Proof. For brevity, in the proof below, we write $X < Y$ if X is ordered before Y under the assumed linear ordering $X_1, ..., X_n$ of the variables in ϕ, and write $X \leq Y$ if $X < Y$

or $X = Y$. Note that X *higher* than Y implies $X < Y$; the converse is *not* necessarily true.

Suppose $\mathcal{T}$ is a successful tableau which contains a μ-trail. Such a μ-trail must contain a subtrail:

$$\tau \quad = \quad (u_1, \psi_1^{\rho_1}) \rightarrow (u_2, \psi_2^{\rho_2}) \rightarrow ...$$

such that each formula $\psi_i^{\rho_i}$ occurs infinitely often on this subtrail. Suppose Z is the *highest* variable unfolded infinitely often in τ. Obviously, Z is a μ-variable and is also the highest variable unfolded in τ (because every variable unfolded in τ is unfolded at infinitely many places). Thus, τ contains *infinitely many* occurrences of subtrails of the form $(u_i, Z^{\rho}) \rightarrow (u_{i+1}, \psi^{(\rho \restriction Z) \cdot z})$. This implies that the list $\rho_1 \restriction Z$, $\rho_2 \restriction Z$, ... does *not* converge, i.e. for each $i \geq 1$ there exists $i' > i$ such that $\rho_i \restriction Z \neq \rho_{i'} \restriction Z$. Let Y be the least μ-variable (w.r.t. $<$) such that the list $\rho_1 \restriction Y$, $\rho_2 \restriction Y$, ... does *not* converge (so Y could be Z). Hence, there is some $j \geq 1$ such that

(1) $\rho_j \restriction Y' = \rho_{j+1} \restriction Y' = ...$, for each μ-variable $Y' < Y$, and
(2) the list $\rho_j \restriction Y, \rho_{j+1} \restriction Y, ...$ does *not* converge.

Let ρ be an element in the list $\rho_j \restriction Y, \rho_{j+1} \restriction Y, ...$ which has the *least* length. We claim that ρ is a prefix of $\rho_i \restriction Y$, for each $i \geq j$. Suppose the length of ρ is n (thus each $\rho_i \restriction Y$, $i \geq j$, has length at least n). We first show that

$$\rho_i \restriction n \succeq_{\Theta_i} \rho_{i+1} \restriction n, \text{ for each } i \geq j,$$

where Θ_i is the global sequence in node u_i. To show this, we consider how the formula $\psi_i^{\rho_i}$ in u_i is reduced to $\psi_{i+1}^{\rho_{i+1}}$ in u_{i+1} (see Definition 4.42). One obvious case is where the tableau rule applied at u_i does *not* affect $\psi_i^{\rho_i}$. Another obvious case is where u_i is a terminal and u_{i+1} is its companion. In both of these cases, $\psi_i^{\rho_i} = \psi_{i+1}^{\rho_{i+1}}$. For other cases below, we suppose a tableau rule R is applied at u_i and reduces the formula $\psi_i^{\rho_i}$ to $\psi_{i+1}^{\rho_{i+1}}$.

- $\mathsf{R} = \mathsf{R}\wedge$. In this case, $\psi_i^{\rho_i} = (\gamma_1 \wedge \gamma_2)^{\rho_i}$ (for some formulae γ_1, γ_2) and $\psi_{i+1}^{\rho_{i+1}} = \gamma_k^{\rho_i}$, for some $k \in \{1, 2\}$. Thus, in this case, $\rho_{i+1} = \rho_i$.
- $\mathsf{R} = \mathsf{R}\vee$, $\mathsf{R}\sigma$, or $\mathsf{R}\langle\rangle$. As in the previous case, we have $\rho_{i+1} = \rho_i$.
- $\mathsf{R} = \mathsf{Unfold}_{\mu}$. In this case, $\psi_i^{\rho_i} = X^{\rho_i}$, for some μ-variable X, and $\psi_{i+1}^{\rho_{i+1}} = \psi^{(\rho_i \restriction X) \cdot x}$, where ψ is the unfolding of X and x is a name for X. Since $\rho_{i+1} \restriction X \neq \rho_i \restriction X$, by condition (1) above, $Y \leq X$. If $Y < X$, then $\rho_{i+1} \restriction Y = ((\rho_i \restriction X) \cdot x) \restriction Y = \rho_i \restriction Y$. If $Y = X$, then, obviously, $\rho_i \restriction Y$ is a proper prefix of $\rho_{i+1} \restriction Y$. In both cases, since the length of $\rho_i \restriction Y$ is $\geq n$, we may deduce that $\rho_i \restriction n = \rho_{i+1} \restriction n$.
- $\mathsf{R} = \mathsf{Unfold}_{\nu}$. In this case, $\psi_i^{\rho_i} = X^{\rho_i}$, for some ν-variable X, and $\psi_{i+1}^{\rho_{i+1}} = \psi^{\rho_i \restriction X}$, where ψ is the unfolding of X. Since Z is the highest variable unfolded in τ, Z must be higher than X, and hence $Z < X$. Since $Y \leq Z$, this implies $Y < X$. Therefore, $\rho_{i+1} \restriction Y = (\rho_i \restriction X) \restriction Y = \rho_i \restriction Y$. Since the length of $\rho_i \restriction Y$ is $\geq n$, we have $\rho_i \restriction n = \rho_{i+1} \restriction n$.

- $\mathsf{R} = \mathsf{Thin}$. In this case, $\psi_i = \psi_{i+1}$ and one of the following holds:
  - $\rho_i = \rho_{i+1}$, or
  - $\rho_i \succ_{\Theta_i} \rho_{i+1}$, or
  - $\rho_i \restriction X$ is a proper prefix of $\rho_{i+1} \restriction X$, for some μ-variable X.

  The first two cases clearly imply that $\rho_i \restriction n \succeq_{\Theta_i} \rho_{i+1} \restriction n$. Suppose the last case holds. Clearly, by (1), $Y \leq X$. This implies that either $\rho_i \restriction Y = \rho_{i+1} \restriction Y$ or $\rho_i \restriction Y$ is a proper prefix of $\rho_{i+1} \restriction Y$. Since the length of $\rho_i \restriction Y$ is $\geq n$, it follows that $\rho_i \restriction n = \rho_{i+1} \restriction n$.
- $\mathsf{R} = \mathsf{Reset}_x$, for some name x. In this case, $\psi_i = \psi_{i+1}$ and ρ_{i+1} is a proper prefix ρ_i. Since the length of $\rho_{i+1} \restriction Y$ is $\geq n$, it follows that $\rho_i \restriction n = \rho_{i+1} \restriction n$.

Therefore, we have $\rho_i \restriction n \succeq_{\Theta_i} \rho_{i+1} \restriction n$, for each $i \geq j$. Since ρ_j occurs infinitely often in the trail, by Lemma 4.43, $\rho_j \restriction n = \rho_{j+1} \restriction n = ...$, which implies that ρ is a prefix of each $\rho_j, \rho_{j+1}, ...$, as claimed.

Since the list $\rho_j \restriction Y, \rho_{j+1} \restriction Y, \rho_{j+2} \restriction Y, ...$ does not converge and ρ occurs infinitely often in this list, there must be *infinitely many* $i \geq j$ such that $\rho_{i+1} \restriction Y = \rho$ is a *proper* prefix of $\rho_i \restriction Y$. It is not difficult to see that this can only happen when Reset_x, where x is the last name in ρ, is applied at u_i. To see this, we first observe that, since $\rho_i \neq \rho_{i+1}$, $\psi_i^{\rho_i}$ must be reduced at u_i by either Unfold_μ, Unfold_ν, Thin, or Reset. We consider each of these possibilities:

- Suppose ψ_i is a μ-variable X and $\psi_{i+1}^{\rho_{i+1}}$ is the unfolding of X^{ρ_i}. Hence, $\rho_{i+1} = (\rho_i \restriction X) \cdot x$, for some name x for X. Since Z is higher than or equal to X, it follows that $Z \leq X$ and hence $Y \leq X$. If $Y < X$, then $\rho_{i+1} \restriction Y = ((\rho_i \restriction X) \cdot x) \restriction Y = \rho_i \restriction Y$. If $Y = X$, then obviously $\rho_i \restriction Y$ is a proper prefix of $\rho_{i+1} \restriction Y$. Both of these contradict the fact that $\rho_{i+1} \restriction Y$ is a proper prefix of $\rho_i \restriction Y$.
- Suppose ψ_i is a ν-variable X and $\psi_{i+1}^{\rho_{i+1}}$ is the unfolding of X^{ρ_i}. Hence, $\rho_{i+1} = \rho_i \restriction X$. Since Z must be higher than X, it follows that $Z < X$ and hence $Y < X$. But this implies that $\rho_{i+1} \restriction Y = (\rho_i \restriction X) \restriction Y = \rho_i \restriction Y$, contradicting the fact that $\rho_{i+1} \restriction Y$ is a proper prefix of $\rho_i \restriction Y$.
- Suppose $\psi_{i+1}^{\rho_{i+1}}$ is reduced from $\psi_i^{\rho_i}$ in u_i by Thin. Hence, $\psi_{i+1} = \psi_i$ and both $\psi_{i+1}^{\rho_{i+1}}$ and $\psi_i^{\rho_i}$ are in u_i. But since $\rho_{i+1} \restriction Y$ is a proper prefix of $\rho_i \restriction Y$, the thinning rule should discard the formula $\psi_{i+1}^{\rho_{i+1}}$ instead of $\psi_i^{\rho_i}$. Thus, this case is not possible.
- Suppose $\psi_{i+1}^{\rho_{i+1}}$ is reduced from $\psi_i^{\rho_i}$ in u_i by Reset_x. Hence, x is the last name of ρ_{i+1} and ρ_{i+1} is a proper prefix of ρ_i. The name x must lie within $\rho_{i+1} \restriction Y$, for otherwise $\rho_{i+1} \restriction Y$ and $\rho_i \restriction Y$ would be equal. Hence x is the last name of $\rho_{i+1} \restriction Y = \rho$.

Therefore, the rule Reset_x, where x is the last name in ρ, is applied at u_i. Since there are infinitely many such i, we may deduce that the rule Reset_x is applied infinitely often on τ. And since ρ is a prefix of each ρ_i ($i \geq j$), the name x appears in each node along

τ. Since the tableau is finite, it follows that there must be a path $v, ..., u$ in $\mathcal{T}$, where u is a leaf and v is its companion, such that Reset_x is applied on this path and the name x occurs throughout the path. This contradicts the assumption that each leaf of $\mathcal{T}$ is successful. Hence $\mathcal{T}$ cannot contain a μ-trail. □

It is not surprising that the non-existence of μ-trails in a tableau for ϕ implies that the model constructed above is indeed a model for ϕ. The proof is very similar to the soundness proof of the tableau system TS_0. The following basic result, which follows easily from the semantics, will be employed in the proof.

Lemma 4.45. *For any model $\mathcal{M}$, valuation $\mathcal{V}$, state s, and formula $\sigma X.\psi$, if $\mathcal{M}, s \models_{\mathcal{V}} \psi$ and $\mathcal{M}, s \not\models_{\mathcal{V}} \sigma X.\psi$, then there must be state $t \in \mathcal{V}(X)$ such that $\mathcal{M}, t \not\models_{\mathcal{V}} \sigma X.\psi$.*

Proof. Suppose $\mathcal{M}, s \models_{\mathcal{V}} \psi$ and $\mathcal{M}, s \not\models_{\mathcal{V}} \sigma X.\psi$. But suppose we assume that, for each $t \in \mathcal{V}(X)$, $\mathcal{M}, t \models_{\mathcal{V}} \sigma X.\psi$ (thus, $\mathcal{V}(X) \subseteq \|\sigma X.\psi\|_{\mathcal{V}}$). From the semantics and the Knaster-Tarski Theorem (Theorem 0.1), $\|\sigma X.\psi\|_{\mathcal{V}}$ is a fixpoint of $\lambda S.\|\psi\|_{\mathcal{V}[X:=S]}$, i.e.

$$\|\psi\|_{\mathcal{V}[X:=\|\sigma X.\psi\|_{\mathcal{V}}]} = \|\sigma X.\psi\|_{\mathcal{V}}.$$

Since, by assumption, $\mathcal{V}(X) \subseteq \|\sigma X.\psi\|_{\mathcal{V}}$, it follows by monotonicity (Proposition 2.7), that

$$\|\psi\|_{\mathcal{V}} \subseteq \|\psi\|_{\mathcal{V}[X:=\|\sigma X.\psi\|_{\mathcal{V}}]} = \|\sigma X.\psi\|_{\mathcal{V}}.$$

Since $\mathcal{M}, s \models_{\mathcal{V}} \psi$, this implies that $\mathcal{M}, s \models_{\mathcal{V}} \sigma X.\psi$, contradicting our assumption. Hence, there must be a state $t \in \mathcal{V}(X)$ such that $\mathcal{M}, t \not\models_{\mathcal{V}} \sigma X.\psi$. □

Lemma 4.46. *If a TS-tableau $\mathcal{T}$ for ϕ does not contain a μ-trail, then $\mathcal{M}_{\mathcal{T}}$ is a model of ϕ.*

Proof. Assume that $\mathcal{T}$ is a tableau for ϕ without a μ-trail. We will prove a more general statement that, for any state s in $\mathcal{M}_{\mathcal{T}}$ and each node $u \in [s]$, if ψ^ρ is in u then ψ is true at s in $\mathcal{M}_{\mathcal{T}}$. To prove this, we first define, for each pair (u, ψ^ρ) where u is a node and ψ^ρ is in the goal at u, a *valuation* $\mathrm{Val}(u, \psi^\rho)$. Namely, for each variable X,

- $\mathrm{Val}(u, \psi^\rho)(X) = \{s \in S \mid$ for some node $v \in [s]$ and formula $X^{\rho'}$ in v, there is a trail from (u, ψ^ρ) to $(v, X^{\rho'})$ along which X is active $\}$.

The following properties are easy to show from the above definition.

(a) $\mathrm{Val}(u, \psi^\rho)(X) = \emptyset$, for any variable X *not* active in ψ.
(b) If there is a trail from (u, ψ^ρ) to $(v, \psi'^{\rho'})$ along which X is active, then $\mathrm{Val}(v, \psi'^{\rho'})(X) \subseteq \mathrm{Val}(u, \psi^\rho)(X)$.
(c) If there is a trail from (u, ψ^ρ) to $(v, \psi'^{\rho'})$ such that each variable which is active in ψ' is active along this trail, then $\mathrm{Val}(u, \psi^\rho)$ extends $\mathrm{Val}(v, \psi'^{\rho'})$[3].

[3] A valuation $\mathcal{V}$ extends a valuation $\mathcal{V}'$ iff $\mathcal{V}'(X) \subseteq \mathcal{V}(X)$ for each variable X.

(d) If there is a trail from (u, ψ^ρ) to $(v, X^{\rho'})$ along which X is active, then $\mathrm{Val}(u, \psi^\rho)$ extends $\mathrm{Val}(v, X^{\rho'})$.

(a) and (b) follows directly from the definition. (c) follows easily from (a) and (b). (d) follows from (c) and the fact that, for any variables X, Y, if X is active along a trail τ and Y is active in X, then Y is also active along τ.

We first prove that the following claim holds for any formula ψ^ρ:

Claim ($\star$) For any state s and any node $u \in [s]$, if ψ^ρ is in u, then $s \models_{\mathrm{Val}(u,\psi^\rho)} \psi$.

Use induction on ψ.

- $\psi = P, \neg P$. Follows from the definition of $\mathcal{V}_{\mathrm{Prop}}$ in $\mathcal{M}_{\mathcal{T}}$.
- $\psi = X$. By definition, if $u \in [s]$ and X^ρ is in u, then $s \in \mathrm{Val}(u, X^\rho)(X)$. Hence $s \models_{\mathrm{Val}(u,X^\rho)} X$.
- $\psi = \psi_1 \wedge \psi_2$. Suppose $(\psi_1 \wedge \psi_2)^\rho$ labels a node $u \in [s]$. There must be some node $u' \in [s]$ such that the formula is reduced via $\mathsf{R}\wedge$, i.e. there is a trail $(u, (\psi_1 \wedge \psi_2)^\rho) \to ... \to (u', (\psi_1 \wedge \psi_2)^{\rho'}) \to (v, \psi_i^{\rho'})$, for each $i \in \{1, 2\}$. By (c), $\mathrm{Val}(u, (\psi_1 \wedge \psi_2)^\rho)$ extends $\mathrm{Val}(v, \psi_i^{\rho'})$, for each i. Applying the induction hypothesis, we have $s \models_{\mathrm{Val}(v,\psi_i^{\rho'})} \psi_i$, which implies that $s \models_{\mathrm{Val}(u,(\psi_1\wedge\psi_2)^\rho)} \psi_i$, for each i. Therefore, $s \models_{\mathrm{Val}(u,(\psi_1\wedge\psi_2)^\rho)} \psi_1 \wedge \psi_2$.
- $\psi = \psi_1 \vee \psi_2$. Similar to the previous case.
- $\psi = \langle a \rangle \psi'$. Suppose $(\langle a \rangle \psi')^\rho$ labels a node $u \in [s]$. This means that $(\langle a \rangle \psi')^\rho$ is reduced at s via $\mathsf{R}\langle\rangle$. Hence, there must be a state $t \in R_a(s)$ and node $v \in [t]$ such that there is a trail $(u, (\langle a \rangle \psi')^\rho) \to ... \to (s, (\langle a \rangle \psi')^{\rho'}) \to (v, \psi'^{\rho'})$. By (c), $\mathrm{Val}(u, (\langle a \rangle \psi')^\rho)$ extends $\mathrm{Val}(v, \psi'^{\rho'})$. Applying the induction hypothesis, $t \models_{\mathrm{Val}(v,\psi'^{\rho'})} \psi'$. Thus, we have $t \models_{\mathrm{Val}(u,(\langle a\rangle\psi')^\rho)} \psi'$. Therefore, $s \models_{\mathrm{Val}(u,(\langle a\rangle\psi')^\rho)} \langle a \rangle \psi'$.
- $\psi = [a]\psi'$. Similar to the previous case.
- $\psi = \mu X.\psi'$. We first prove a more general claim:

  for any state s and node $u \in [s]$, if X^ρ is in u then $s \models_{\mathrm{Val}(u,X^\rho)} \mu X.\psi'$.

  Suppose there is a state s and a node $u \in [s]$ such that the claim fails, i.e. X^ρ is in u but $s \not\models_{\mathrm{Val}(u,X^\rho)} \mu X.\psi'$. Since X^ρ is in u, there must be a trail from (u, X^ρ) leading to the unfolding of X, i.e. $(u, X^\rho) \to ... \to (u', \psi'^{\rho'})$, for some $u' \in [s]$. Applying the induction hypothesis, we have

  $$s \models_{\mathrm{Val}(u',\psi'^{\rho'})} \psi'. \tag{4.1}$$

  Since $\mathrm{Val}(u, X^\rho)$ extends $\mathrm{Val}(u', \psi'^{\rho'})$ and $s \not\models_{\mathrm{Val}(u,X^\rho)} \mu X.\psi'$, it follows that

  $$s \not\models_{\mathrm{Val}(u',\psi'^{\rho'})} \mu X.\psi'. \tag{4.2}$$

  From (4.1) and (4.2) above and Lemma 4.45, it follows that there is a state $t \in \mathrm{Val}(u', \psi'^{\rho'})(X)$ such that $t \not\models_{\mathrm{Val}(u',\psi'^{\rho'})} \mu X.\psi'$. By definition, $t \in \mathrm{Val}(u', \psi'^{\rho'})(X)$

implies that, for some node $v \in [t]$ and formula $X^{\rho''}$ in v, there is a trail from $(u', \psi'^{\rho'})$ to $(v, X^{\rho''})$ along which X is active. Hence, by (d), $\mathrm{Val}(u', \psi'^{\rho'})$ extends $\mathrm{Val}(v, X^{\rho''})$. Therefore, $t \not\models_{\mathrm{Val}(v,X^{\rho''})} \mu X.\psi'$. So the claim also fails for state t and node $v \in [t]$. Moreover, since there is a trail from (u, X^{ρ}) to $(u', \psi'^{\rho'})$ and a trail from $(u', \psi'^{\rho'})$ to $(v, X^{\rho''})$ where X is active in both of these trails, this clearly implies that there is a trail from (u, X^{ρ}) to $(v, X^{\rho''})$ in which X is active and unfolded. We may then repeat the above argument using state t and node v, and so on, to obtain an infinite trail in which X is active and unfolded infinitely often. This contradicts the assumption that $\mathcal{T}$ does not contain a μ-trail. Hence, the above claim must hold.

Back to the main proof, suppose $(\mu X.\psi')^{\rho}$ is in node $u \in [s]$. There must be some node $u' \in [s]$ such that the formula is reduced via $\mathsf{R}\mu$, i.e. there is a trail $(u, (\mu X.\psi')^{\rho}) \to ... \to (u', (\mu X.\psi')^{\rho'}) \to (v, X^{\rho'})$. This implies that $\mathrm{Val}(v, X^{\rho'})(Y) \subseteq \mathrm{Val}(u, (\mu X.\psi')^{\rho})(Y)$ for each variable Y higher than X. By what we have just shown, $s \models_{\mathrm{Val}(v,X^{\rho'})} \mu X.\psi'$. This implies that $s \models_{\mathrm{Val}(u,(\mu X.\psi')^{\rho})} \mu X.\psi'$.

- $\psi = \nu X.\psi'$. We first claim that:

  for any state s and node $u \in [s]$, if X^{ρ} is in u, then $s \models_{\mathrm{Val}(u,X^{\rho})} \nu X.\psi'$.

  Suppose t is any state in $\mathrm{Val}(u, X^{\rho})(X)$. By definition, for some node $v \in [t]$ and formula $X^{\rho'}$ in v, there is a trail from (u, X^{ρ}) to $(v, X^{\rho'})$ in which X is active. There must be a trail from $(v, X^{\rho'})$ leading to an unfolding of X, i.e. $(v, X^{\rho'}) \to ... \to (v', \psi'^{\rho''})$ for some $v' \in [t]$. Thus, there is a trail from (u, X^{ρ}) to $(v', \psi'^{\rho''})$ in which X is active. By (c), $\mathrm{Val}(u, X^{\rho})$ extends $\mathrm{Val}(v', \psi'^{\rho''})$. Applying the induction hypothesis, we have $t \models_{\mathrm{Val}(v',\psi'^{\rho''})} \psi'$ and, consequently, $t \models_{\mathrm{Val}(u,X^{\rho})} \psi'$. Since this holds for any state $t \in \mathrm{Val}(u, X^{\rho})(X)$, we may conclude that

  $$\mathrm{Val}(u, X^{\rho})(X) \quad \subseteq \quad \|\psi'\|_{\mathrm{Val}(u,X^{\rho})}. \tag{4.3}$$

  From the semantics, we may deduce that

  $$\mathrm{Val}(u, X^{\rho})(X) \quad \subseteq \quad \|\nu X.\psi'\|_{\mathrm{Val}(u,X^{\rho})}. \tag{4.4}$$

  Suppose s is any state such that $u \in [s]$ and X^{ρ} is in u. By definition, $s \in \mathrm{Val}(u, X^{\rho})(X)$. From (4.4), we have $s \models_{\mathrm{Val}(u,X^{\rho})} \nu X.\psi'$, as claimed.

  Back to the main proof, suppose $(\nu X.\psi')^{\rho}$ is in node $u \in [s]$. There must be some node $u' \in [s]$ where the formula is reduced via $\mathsf{R}\nu$, i.e. there is a trail $(u, (\nu X.\psi')^{\rho}) \to ... \to (u', (\nu X.\psi')^{\rho'}) \to (v, X^{\rho'})$. This implies that $\mathrm{Val}(v, X^{\rho'})(Y) \subseteq \mathrm{Val}(u, (\nu X.\psi')^{\rho})(Y)$ for each variable Y higher than X. By what we have just shown, $s \models_{\mathrm{Val}(v,X^{\rho'})} \nu X.\psi'$. Therefore, $s \models_{\mathrm{Val}(u,(\nu X.\psi')^{\rho})} \nu X.\psi'$.

Since ϕ labels the root $u_0 \in [s_0]$ and ϕ is closed, it follows from $(\star)$ that $s_0 \models \phi$. Hence, $\mathcal{M}_{\mathcal{T}}$ is a model for ϕ. □

Theorem 4.47 (Soundness of TS)**.** *Every guarded and closed formula which has a successful TS-tableau has a model in which the number of states is linear in the number of nodes in the tableau.*

Proof. Suppose $\mathcal{T}$ is a successful TS-tableau for the given formula ϕ. By Lemma 4.44, there is no μ-trail in $\mathcal{T}$. By Lemma 4.46, $\mathcal{M}_{\mathcal{T}}$ is a model for ϕ. The model $\mathcal{M}_{\mathcal{T}}$ contains the modal nodes of $\mathcal{T}$ as its states; hence the size of $\mathcal{M}_{\mathcal{T}}$ is clearly linear in the number of nodes in $\mathcal{T}$. □

4.3.2 Completeness

The completeness of TS can be shown along the same line as for tableau system ACON. The key is to define the *signature of a goal*, $\mathrm{Sig}(\Theta \vdash \Gamma)$, as a measure to guarantee the success of the constructed tableau. In particular, we need a notion of signatures that satisfies the properties similar to Lemma 4.27.

As before, by a *name signature*, we mean a function which assigns an ordinal to each name. Name signatures are ordered with respect to a global sequence as given in Definition 4.23, i.e. for any global sequence $\Theta = z_1 ... z_n$,

- $\sigma \approx_\Theta \sigma'$ iff $\sigma(z_i) = \sigma'(z_i)$ for each i.
- $\sigma \prec_\Theta \sigma'$ iff $\sigma(z_j) < \sigma'(z_j)$ for some j and $\sigma(z_i) = \sigma'(z_i)$ for each $i < j$.
- $\sigma \preceq_\Theta \sigma'$ iff $\sigma \prec_\Theta \sigma'$ or $\sigma \approx_\Theta \sigma'$.

The notion of a *good* name signature for a set Γ of formulae needs to be generalised, as a sequence ρ in Γ may contain more than one name for the same μ-variable. First, for each name signature σ and sequence ρ, we define the signature σ_ρ as follows: for each μ-variable Z,

- if ρ contains a name for Z, then $\sigma_\rho(Z) = \sigma(z_i)$ where z_i is the name for Z occurring *last* in ρ;
- otherwise, $\sigma_\rho(Z) = \omega_1$ where ω_1 denotes the least uncountable ordinal.[4]

Definition 4.48. A name signature σ is considered *good* for Γ iff

G1. for any sequence ρ occurring in Γ and names z_i, z_j for the same variable, if z_i occurs before z_j in ρ, then $\sigma(z_i) > \sigma(z_j)$;[5] and

G2. there is a countable model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \psi$, for each $\psi^\rho \in \Gamma$.

Condition G1 is introduced for technical reason. It should become clear later in the proof. It is quite obvious that every satisfiable set Γ has a good name signature.

[4] In fact, we may choose any ordinal greater than every countable ordinal instead of ω_1. An alternative solution is to extend $\mathbb{O}$ with a top element ∞ which is greater than every ordinal, and use ∞ instead of ω_1.

[5] Notice that if a name signature σ satisfies G1, for any sequence ρ in Γ and μ-variable Z, $\sigma_\rho(Z)$ is the least ordinal in $\{\sigma(z_i) \mid z_i \text{ is a name for } Z \text{ in } \rho\}$ if ρ contains a name for Z, otherwise $\sigma_\rho(Z) = \omega_1$.

Lemma 4.49. *For any goal* $\Theta \vdash \Gamma$*, if* Γ *is satisfiable, then there is a* good *name signature for* Γ*.*

Proof. If Γ is satisfiable (and finite), there must be a countable model $\mathcal{M}$, a state s, and a least ordinal α such that $\mathcal{M}, s \models_{\langle \alpha,\ldots,\alpha\rangle} \psi$ for each formula ψ in Γ.

Suppose $\Theta = z_1 ... z_n$. Define a name signature σ:

- $\sigma(z_i) = \alpha + n - i$ for each name z_i, and
- $\sigma(x) = 0$ for each name x *not* occurring in Θ.

It is easy to check that σ is a good name signature for Γ. Condition G1 holds because, for any sequence ρ occurring in Γ, if a name z_i occurs before z_j in ρ, then z_i must also occurs before z_j in Θ which means that $\sigma(z_i) > \sigma(z_j)$. G2 holds because, for any ρ occurring in Γ, each ordinal in σ_ρ is no less than α, and hence $\mathcal{M}, s \models_{\sigma_\rho} \psi$ for each ψ^ρ in Γ. □

The definition of the signature $\mathrm{Sig}(\Theta \vdash \Gamma)$ of a goal $\Theta \vdash \Gamma$, where Γ is satisfiable, can be given as in Definition 4.25, i.e. $\mathrm{Sig}(\Theta \vdash \Gamma)$ is the name signature σ such that

- σ is good for Γ,
- $\sigma \preceq_\Theta \sigma'$ for any good name signature σ' for Γ, and
- $\sigma(z) = 0$ for each name z *not* occurring in Θ.

As in the completeness proof of ACON, the following properties of signatures are the key to the completeness proof.

Lemma 4.50. *Below* Θ' *denotes the result of removing all the names in* Θ *not occurring in any augmented formula in the goal on the right hand side.*

(a) $\Gamma' \subseteq \Gamma$ *implies* $\mathrm{Sig}(\Theta \vdash \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \Gamma')$.

(b) $\mathrm{Sig}(\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma)$.

(c) $\mathrm{Sig}(\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_i^\rho, \Gamma)$ *for some* $i \in \{1, 2\}$.

(d) $\mathrm{Sig}(\Theta \vdash (\mu Z.\psi)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma)$.

(e) $\mathrm{Sig}(\Theta \vdash (\nu X.\psi)^\rho, \Gamma) \succeq_\Theta \mathrm{Sig}(\Theta \vdash X^\rho, \Gamma)$.

(f) $\mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \cdot z^i \vdash \psi^{(\rho|Z)\cdot z^i}, \Gamma)$ *where* Z *identifies* $\mu Z.\psi$ *and* z^i *is a name for* Z not *occurring in* Θ.

(g) $\mathrm{Sig}(\Theta \vdash X^\rho, \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^{\rho|X}, \Gamma)$ *where* X *identifies* $\nu X.\psi$.

(h) $\mathrm{Sig}(\Theta \vdash (\langle a\rangle\psi)^\rho, \Gamma) \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^\rho, \Gamma_a)$ *where* $\Gamma_a = \{\gamma^{\rho'} \mid ([a]\gamma)^{\rho'} \in \Gamma\}$.

Proof. We describe case (c), (d), (f), and (h) only. Other cases are similar.

(c) Let σ be $\mathrm{Sig}(\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma)$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \psi_1 \vee \psi_2$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. This implies that $\mathcal{M}, s \models_{\sigma_\rho} \psi_i$ for some i. Hence σ is good for $\{\psi_i^\rho\} \cup \Gamma$, which implies that $\sigma \succeq_\Theta \mathrm{Sig}(\Theta \vdash \psi_i^\rho, \Gamma)$.

(d) Let σ be $\mathrm{Sig}(\Theta \vdash (\mu Z.\psi)^\rho, \Gamma)$. Thus there is a countable model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \mu Z.\psi$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. Clearly, ρ does not contain a name for Z. Hence, by definition, $\sigma_\rho(Z) = \omega_1$. Since $\mathcal{M}$ is a countable model, $\mathcal{M}, s \models_{\sigma_\rho} \mu^{\omega_1} Z.\psi$, and so $\mathcal{M}, s \models_{\sigma_\rho} Z$. Thus σ is a good signature for $\{Z^\rho\} \cup \Gamma$, which implies that $\sigma \succeq_\Theta \mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma)$.

(f) Let σ be $\mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma)$, where Z identifies $\mu Z.\psi$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} Z$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. Since any μ-variable Z' lower than Z is *not* active in Z, we have $\mathcal{M}, s \models_{\sigma_{\rho\restriction Z}} Z$. This implies that there is an ordinal $\alpha < \sigma_\rho(Z)$ such that

$$\mathcal{M}, s \models_{\sigma_{\rho\restriction Z}} \psi\{\mu^\alpha Z.\psi/Z\}, \tag{4.5}$$

Let z^i be a name for Z *not* occurring in Θ. Consider the set

$$\{\psi^{(\rho\restriction Z)\cdot z^i}\} \cup \Gamma.$$

Let σ' be $\sigma[z^i := \alpha]$. It can be shown that σ' is good for the above set, i.e. satisfying conditions G1 and G2. To check that G1 holds, we need to show that for each ρ' occurring in the above set

- for any names z_j, z_k in ρ' for the same variable, if z_j occurs before z_k in ρ', then $\sigma'(z_j) > \sigma'(z_k)$.

Since σ is good for $\{Z^\rho\} \cup \Gamma$, this is obviously the case for any ρ' *not* containing the new name z^i. But this holds for $(\rho \restriction Z) \cdot z^i$ too because $\sigma'(z^i) = \alpha < \sigma(z^j)$ for any name z^j for Z in ρ. Hence condition G1 holds. For G2, since $\sigma'_{\rho'} = \sigma_{\rho'}$ for each ρ' in Γ, we have $\mathcal{M}, s \models_{\sigma'_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. And since $\sigma'_{(\rho\restriction Z)\cdot z^i}(Z) = \sigma'(z_i) = \alpha$ and $\sigma'_{(\rho\restriction Z)\cdot z^i}(Z') = \sigma_{\rho\restriction Z}(Z')$ for each $Z' \neq Z$, it follows from (4.5) that

$$\mathcal{M}, s \models_{\sigma'_{(\rho\restriction Z)\cdot z^i}} \psi. \tag{4.6}$$

Thus σ' is a good name signature for $\{\psi^{(\rho\restriction Z)\cdot z_i}\} \cup \Gamma$, and hence

$$\sigma' \quad \succeq_{\Theta'\cdot z^i} \quad \mathrm{Sig}(\Theta' \cdot z^i \vdash \psi^{(\rho\restriction Z)\cdot z^i}, \Gamma).$$

Since σ and σ' agree everywhere except at z^i, we have $\sigma \succeq_{\Theta'} \mathrm{Sig}(\Theta' \cdot z^i \vdash \psi^{(\rho\restriction Z)\cdot z^i}, \Gamma)$, as required.

(h) Let σ be $\mathrm{Sig}(\Theta \vdash (\langle a\rangle\psi)^\rho, \Gamma)$. Thus there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_\rho} \langle a\rangle\psi$ and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. This means that there is a state $t \in R_a(s)$ such that $\mathcal{M}, t \models_{\sigma_\rho} \psi$ and $\mathcal{M}, t \models_{\sigma_{\rho'}} \gamma$ for each $([a]\gamma)^{\rho'} \in \Gamma$.

Hence σ is a good signature for $\{\psi^\rho\} \cup \Gamma_a$. Thus $\sigma \succeq_{\Theta'} \mathrm{Sig}(\Theta' \vdash \psi^\rho, \Gamma_a)$.

□

Lemma 4.51. *Suppose* $\Theta \vdash \psi_1^{\rho \cdot z \cdot z_1 \cdot \rho_1}, ..., \psi_n^{\rho \cdot z \cdot z_n \cdot \rho_n}, \Gamma$ *is a goal where* $z, z_1, ..., z_n$ *are names for the* same *variable, and* z *does* not *occur in* Γ.

$$\mathrm{Sig}(\Theta \vdash \psi_1^{\rho \cdot z \cdot z_1 \cdot \rho_1}, ..., \psi_n^{\rho \cdot z \cdot z_n \cdot \rho_n}, \Gamma) \quad \succ_{\Theta''} \quad \mathrm{Sig}(\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma),$$

where Θ' *is* Θ *with all the names* not *occurring in the latter goal removed, and* Θ'' *is any prefix of* Θ' *which contains* z.

Proof. Suppose $z, z_1, ..., z_n$ are names for Z. For brevity, let ρ_i' denote $\rho \cdot z \cdot z_i \cdot \rho_i$. Let σ be $\mathrm{Sig}(\Theta \vdash \psi_1^{\rho_1'}, ..., \psi_n^{\rho_n'}, \Gamma)$. By condition G1, $\sigma(z) > \sigma(z_i)$ for each i. Let α denote the greatest ordinal in $\{\sigma(z_1), ..., \sigma(z_n)\}$. Hence $\sigma(z) > \alpha$.

Let σ' be $\sigma[z := \alpha]$. It can be shown that σ' is a good name signature for $\{\psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}\} \cup \Gamma$. We first check that G1 holds. Since z does *not* occur in Γ, we only need to check that $\sigma'(z') > \sigma'(z)$ for each name z' for Z occurring in ρ. But this is the case because $\sigma'(z') = \sigma(z') > \sigma(z) > \alpha = \sigma'(z)$.

For G2, we know that there is a model $\mathcal{M}$ and state s such that $\mathcal{M}, s \models_{\sigma_{\rho_i'}} \psi_i$ for each i and $\mathcal{M}, s \models_{\sigma_{\rho'}} \gamma$ for each $\gamma^{\rho'} \in \Gamma$. The latter implies that $\mathcal{M}, s \models_{\sigma'_{\rho'}} \gamma$ (for each $\gamma^{\rho'} \in \Gamma$), because z does *not* occur in any ρ' in Γ. To show that $\mathcal{M}, s \models_{\sigma'_{\rho \cdot z}} \psi_i$, we consider the values of $\sigma_{\rho_i'}(Z')$ and $\sigma'_{\rho \cdot z}(Z')$ for each μ-variable Z':

(1) [Z' higher than Z] Each name for Z' occurring in $\rho_i' = \rho \cdot z \cdot z_i \cdot \rho_i$ must occur within ρ. Hence the names for Z' in $\rho \cdot z$ are precisely those in ρ_i'. This implies that $\sigma'_{\rho \cdot z}(Z') = \sigma_{\rho_i'}(Z')$.
(2) [$Z' = Z$] $\sigma'_{\rho \cdot z}(Z) = \sigma'(z) = \alpha \geq \sigma(z_i) \geq \sigma_{\rho_i'}(Z)$.
(3) [Otherwise] There is no name for Z' in $\rho \cdot z$. Hence $\sigma'_{\rho \cdot z}(Z') = \omega_1 \geq \sigma_{\rho_i'}(Z')$.

In any of these cases, $\sigma'_{\rho \cdot z}(Z') \geq \sigma_{\rho_i'}(Z')$ for each μ-variable Z'. Therefore $\mathcal{M}, s \models_{\sigma'_{\rho \cdot z}} \psi_i$ for each i. Thus σ' is good for $\{\psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}\} \cup \Gamma$, which implies that

$$\sigma' \quad \succeq_{\Theta'} \quad \mathrm{Sig}(\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma).$$

Since $\sigma' = \sigma[z := \alpha]$ and $\sigma(z) > \alpha$, we have $\sigma \succ_{\Theta''} \sigma'$ for any prefix Θ'' of Θ' which contains z. Hence

$$\sigma \quad \succ_{\Theta''} \quad \mathrm{Sig}(\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma).$$

□

We are now ready to prove the completeness of TS. The tableau that we are constructing will have some uniformity which will later enable us to prove the small model property. We call such a tableau a *uniform tableau*.

Definition 4.52 (Uniform Tableaux)**.** A tableau is said to be *uniform* iff, for any pair of *non-terminal* nodes u, v with the same goal, the tableau rule applied at u is the same as the one applied at v, and the goals of the children of u are the same as those of the children of v.

Theorem 4.53 (Completeness of TS)**.** *Every satisfiable closed formula has a successful and uniform TS-tableau.*

Proof. Suppose ϕ is a satisfiable and closed formula. The construction of a successful tableau for ϕ starts with the smallest tableau $\mathcal{T}_0$ with only the initial goal $\vdash \phi$. We subsequently expand $\mathcal{T}_0$ while making sure the set of the formulae in each goal is satisfiable (the initial formula ϕ is satisfiable by assumption). To ensure that the constructed tableaux are uniform, we assume a *selection rule* which, given a goal, specifies which formulae in the goal should be reduced first. Priority should be given to the formulae which are reducible via rule Thin or Reset.

Suppose we have so far constructed $\mathcal{T}_0, ..., \mathcal{T}_i$. For each non-terminal leaf $u : \Theta \vdash \Gamma$ in $\mathcal{T}_i$, apply the tableau rule following to the assumed selection rule. We consider each possible case (here the underlined formulae are those specified by the selection rule).

- $\Gamma = \underline{\psi^{\rho_1}, \psi^{\rho_2}}, \Gamma'$. Apply Thin to create the subgoal $\Theta' \vdash \psi^{\rho_i}, \Gamma'$, for some $i \in \{1, 2\}$. By Lemma 4.50(a),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta'} \quad \mathrm{Sig}(\Theta' \vdash \psi^{\rho_i}, \Gamma').$$

- $\Gamma = \underline{\psi_1^{\rho \cdot z \cdot z_1 \cdot \rho_1}, ..., \psi_n^{\rho \cdot z \cdot z_n \cdot \rho_n}}, \Gamma'$ where $z, z_1, ..., z_n$ are names for the same variable, and z does *not* occur in Γ'. Apply Reset_z to create the subgoal $\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma'$. By Lemma 4.51,

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succ_{\Theta''} \quad \mathrm{Sig}(\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma),$$

  for any prefix Θ'' of Θ' containing z.
- $\Gamma = \underline{(\psi_1 \wedge \psi_2)^\rho}, \Gamma'$. Apply $\mathsf{R}\wedge$ to create the subgoal $\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma'$. By Lemma 4.50(b),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta} \quad \mathrm{Sig}(\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma').$$

- $\Gamma = \underline{(\psi_1 \vee \psi_2)^\rho}, \Gamma'$. Rule $\mathsf{R}\vee$ can be applied to create either $\Theta \vdash \psi_1^\rho, \Gamma'$ or $\Theta \vdash \psi_2^\rho, \Gamma'$. By Lemma 4.50(c), there is a *least* i such that ψ_i, Γ' is satisfiable and

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta} \quad \mathrm{Sig}(\Theta \vdash \psi_i^\rho, \Gamma').$$

  Apply $\mathsf{R}\vee$ to create the i-th subgoal.
- $\Gamma = \underline{(\mu Z.\psi)^\rho}, \Gamma'$. Apply $\mathsf{R}\mu$ to create the subgoal $\Theta \vdash Z^\rho, \Gamma'$. By Lemma 4.50(d),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta} \quad \mathrm{Sig}(\Theta \vdash Z^\rho, \Gamma').$$

- $\Gamma = \underline{(\nu X.\psi)^\rho}, \Gamma'$. Apply $\mathsf{R}\nu$ to create the subgoal $\Theta \vdash X^\rho, \Gamma'$. By Lemma 4.50(e),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_\Theta \quad \mathrm{Sig}(\Theta \vdash X^\rho, \Gamma').$$

- $\Gamma = \underline{Z^\rho}, \Gamma'$. Apply Unfold_μ to create the subgoal $\Theta' \cdot z^i \vdash \psi^{(\rho[Z)\cdot z^i}, \Gamma'$ where z^i is the first name for Z *not* occurring in Θ. By Lemma 4.50(f),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta'} \quad \mathrm{Sig}(\Theta' \cdot z^i \vdash \psi^{(\rho[Z)\cdot z^i}, \Gamma').$$

- $\Gamma = \underline{X^\rho}, \Gamma'$. Apply Unfold_ν to create the subgoal $\Theta' \vdash \psi^{\rho[X}, \Gamma'$. By Lemma 4.50(g),

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta'} \quad \mathrm{Sig}(\Theta' \vdash \psi^{\rho[X}, \Gamma').$$

- $\Gamma = \underline{(\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma'}$ where $n \geq 1$ and Γ' contains only literals and/or $[\cdot]$-formulae. Apply $\mathsf{R}\langle\rangle$ to create n subgoals $\Theta_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}$ $(1 \leq i \leq n)$. By Lemma 4.50(h), each of these subgoals is satisfiable and

$$\mathrm{Sig}(\Theta \vdash \Gamma) \quad \succeq_{\Theta_i} \quad \mathrm{Sig}(\Theta_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}).$$

The constructed tableaux are clearly uniform because each goal is expanded in a unique way. By Lemma 4.41, the construction must terminate at some tableau $\mathcal{T}'$ all whose leaves are terminal. Since, by the above construction, each goal in $\mathcal{T}'$ is satisfiable, there is no goal which contains a complementary pair of literals. Hence all the leaves which contain only literals and/or $[\cdot]$-formulae are successful. It can be shown that other leaves in $\mathcal{T}'$ are also successful. Suppose $u_1 : \Theta_1 \vdash \Gamma_1, ..., u_n : \Theta_n \vdash \Gamma_n$ is the path to a terminal u_n from its companion u_1 (hence $\Theta_1 = \Theta_n$ and $\Gamma_1 = \Gamma_n$). Assume that u_n is unsuccessful. Thus there is some name z such that

- z occurs in each Θ_i, and
- rule Reset_z is applied at some u_j, $1 \leq j < n$.

Suppose $\Theta = z_1 ... z_k$, where $z_k = z$, is the prefix of Θ_1 up to the occurrence of z. Since $\Theta_1 = \Theta_n$, each z_i must also occur throughout the path, for if z_i is removed at some point, z_i cannot occur before z_k in Θ_n (this follows from the observation that no tableau rule swaps the ordering of names in a global sequence and, when a new name is introduced to a goal, it is added to the end of the global sequence). For the same reason, no name other than $z_1, ..., z_{k-1}$ may occur before z_k in each Θ_i on the path. This means that Θ is a prefix of each Θ_i. It follows from what we note in the construction above (and Fact 4.26) that

$$\mathrm{Sig}(\Theta_1 \vdash \Gamma_1) \succeq_\Theta ... \succeq_\Theta \mathrm{Sig}(\Theta_n \vdash \Gamma_n).$$

Since Reset_z is applied at u_j, by Lemma 4.51,

$$\mathrm{Sig}(\Theta_j \vdash \Gamma_j) \quad \succ_\Theta \quad \mathrm{Sig}(\Theta_{j+1} \vdash \Gamma_{j+1}).$$

This is impossible because $\Theta_1 = \Theta_n$ and $\Gamma_1 = \Gamma_n$. Therefore u_n must be successful. Hence every terminal in $\mathcal{T}'$ is successful. $\mathcal{T}'$ is thus a successful and uniform tableau for ϕ. □

4.3.3 Small Model Property

Since every tableau is bounded in size, we can easily deduce the small model theorem from the soundness and completeness of TS previously shown.

Theorem 4.54 (Small Model Theorem)**.** *Every satisfiable guarded formula ϕ has a finite model with $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$ states.*

Proof. By completeness, every satisfiable guarded formula ϕ has a successful and uniform tableau $\mathcal{T}$ in TS. From the soundness proof, the model $\mathcal{M}_\mathcal{T}$ (see Definition 4.14) satisfies ϕ. $\mathcal{M}_\mathcal{T}$ is a finite tree-with-backedges structure. It can be shown that since $\mathcal{T}$ is uniform, $\mathcal{M}_\mathcal{T}$ can be turned into a small model by identifying all the states which correspond to the modal nodes with the same goal.

Lemma 4.55. *Suppose $\mathcal{T}$ is a* uniform *tableau in TS. For any states s, t in $\mathcal{M}_\mathcal{T}$ (i.e. s and t are some modal nodes in $\mathcal{T}$), if the goals at s and t are the same, then s and t are bisimilar in $\mathcal{M}_\mathcal{T}$.*

Proof. Define a relation B over states of $\mathcal{M}_\mathcal{T}$: sBt iff s and t have the same goal in $\mathcal{T}$. We only need to show that B is a bisimulation. Suppose s and t are states such that sBt. From the definition of $\mathcal{M}_\mathcal{T}$, the valuations of each proposition letter at s and t are the same. Suppose sR_as' (for any action a). Since $\mathcal{T}$ is uniform (and the goals at s and t are the same), it is clear from the definition of R_a that there must be a state t' such that tR_at' and the goals at s' and t' are the same. Hence $s'Bt'$. Similarly, if tR_at' then there will be a state s' such that sR_as' and $s'Bt'$. Thus B is a bisimulation, and the lemma follows. □

Let $\mathcal{M}$ be the bisimulation quotient of $\mathcal{M}_\mathcal{T}$. By the above lemma, the number of states in $\mathcal{M}$ is at most the number of possible goals at the modal nodes in $\mathcal{T}$. By Lemma 4.41, this is bounded by $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$. Since model satisfaction is preserved under taking bisimulation quotient, $\mathcal{M}$ is still a model of ϕ. Hence ϕ has a model with $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$ states as required. □

4.3.4 Complexity

Since every TS-tableau for ϕ is finite and bounded by some function on ϕ, it is decidable to determine whether ϕ has a successful tableau in TS. Hence, by the soundness and

completeness of TS, the satisfiability problem for guarded formulae is *decidable*. It is also possible to determine the complexity of the problem from this tableau method. Below, we present a simple *nondeterministic* algorithm which determines whether ϕ has a successful uniform tableau. The algorithm runs in time $2^{O(|\mu\mathrm{Var}(\phi)||\phi|\log(|\phi|))}$. Thus, we obtain the NEXPTIME bound on checking the satisfiability of guarded formulae. It must be noted that this is *not* optimal as the satisfiability problem for the modal μ-calculus is known to be EXPTIME-complete (the EXPTIME upper bound follows from the results in [SE89], [EJ88], [Saf88], while the lower bound follows easily from the EXPTIME-completeness of PDL [FL79]). However, we believe that there is a *deterministic* algorithm which finds a successful tableau for ϕ in exponential time.

Note that, due to the guardedness assumption in the tableau system, our complexity result is restricted to guarded formulae. As mentioned earlier, an unguarded formula can be effectively converted into a guarded one (see Proposition 2.18). However, as we mentioned, there is an exponential blow-up in the conversion. Hence to obtain a stronger complexity result, the tableau system should be modified to cope with unguarded formulae directly. This can be done as explained in Section 3.4.3.

Theorem 4.56. *The satisfiability problem for guarded formulae is in NEXPTIME.*

Proof. Suppose ϕ is a guarded formula. Without loss of generality, we assume that ϕ is closed. We construct a nondeterministic exponential-time algorithm which determines whether ϕ has a successful tableau in TS.

As in the completeness proof of TS, we assume a *selection rule* which, given a goal $\Theta \vdash \Gamma$, specifies which formulae in Γ should be reduced first (obeying the restriction that rules Thin and Reset should be applied whenever possible). Construct a graph $\mathcal{G} = \langle V, E\rangle$ as follows:

- the set V contains all the possible goals in TS-tableaux for ϕ,
- $(\Theta \vdash \Gamma, \Theta' \vdash \Gamma') \in E$ iff the goal $\Theta \vdash \Gamma$ is reduced to $\Theta' \vdash \Gamma'$ according to the selection rule.

By Lemma 4.40, $|V|$ is bounded by $2^{O(|\phi||\mu\mathrm{Var}(\phi)|\log(|\phi|))}$. Call a goal $\Theta \vdash \Gamma$ in V a *R∨-goal* if, according to the selection rule, rule R∨ is to be applied first; similarly, call it a *R⟨⟩-goal* if rule R⟨⟩ is applicable. Hence $\mathcal{G}$ can be seen as a finite game graph with R∨-goals and R⟨⟩-goals as choice nodes for player I and II respectively. A *play* is a *finite* sequence $\pi = \Theta_0 \vdash \Gamma_0, ..., \Theta_n \vdash \Gamma_n$ where

- $\Theta_0 \vdash \Gamma_0$ is the initial goal for ϕ,
- $(\Theta_i \vdash \Gamma_i, \Theta_{i+1} \vdash \Gamma_{i+1}) \in E$, for each $i < n$, and
- either Γ_n contains only literals and $[\cdot]$-formulae or $\Theta_n \vdash \Gamma_n$ is the first goal in π such that there is some $m < n$ where $\Theta_m \vdash \Gamma_m = \Theta_n \vdash \Gamma_n$.

Hence, a play can be seen as a branch in a fully-expanded tableau for ϕ. Player I *wins* such play π iff *one* of the following holds:

(1) The last goal $\Theta_n \vdash \Gamma_n$ is a consistent set of literals and $[\cdot]$-formulae.

(2) The play contains a subsequence $\Theta_m \vdash \Gamma_m, ..., \Theta_n \vdash \Gamma_n$ where $m < n$ and $\Theta_m \vdash \Gamma_m = \Theta_n \vdash \Gamma_n$ such that, for each name z, if rule Reset_z is applied at some goal on this subsequence, then there is a goal on the subsequence where z does not occur.

Player II *wins* π otherwise. A memoryless strategy for player I (II) is a function Σ which, for each $\mathsf{R}\vee$-goal ($\mathsf{R}\langle\rangle$-goal, respectively), chooses one successor in $\mathcal{G}$. Σ is a winning strategy for player I (II) iff player I (II) wins all the plays in which the player makes choices according to Σ. It is easy to see that player I has a memoryless winning strategy iff ϕ has a uniform successful tableau under the assumed selection rule, which by the completeness of TS (Theorem 4.53), is the case iff ϕ is satisfiable. A nondeterministic algorithm first guesses a strategy for player I and then checks whether it is a winning strategy. The latter task can be carried out (deterministically) in time $O(|\phi||\mu\mathrm{Var}(\phi)||V|)$ (this is done by verifying winning condition (2) for each name occurring the graph; by Lemma 4.39, the number of names used is no greater than $|\phi||\mu\mathrm{Var}(\phi)|$). This means that there is a nondeterministic algorithm which determines whether ϕ is satisfiable in time $O(|\phi||\mu\mathrm{Var}(\phi)||V|) = 2^{O(|\phi||\mu\mathrm{Var}(\phi)|\log(|\phi|))}$. □

4.3.5 Relation to Safra Construction

In this section, we explain in more detail how Safra's determinisation construction [Saf88] is related to our tableau system TS. As mentioned in [NW97], the existence of a successful tableau in TS_0 can be determined using automata. The idea is that, given a formula ϕ, we can effectively construct an infinite-tree automaton recognising the successful TS_0-tableaux for ϕ. Using the method described in [SE89], such an automaton can be obtained by taking a product of two automata: the *local automaton*, which is a simple infinite-tree automaton recognising the (fully-expanded) TS_0-tableaux for ϕ in which every goal does not contain complementary literals, and the *global automaton*, which is a deterministic infinite-word automaton that runs on each infinite branch of a TS_0-tableau to check that there is no μ-trail on the branch. To construct a global automaton, it is simpler to first construct a (nondeterministic) automaton which does the opposite, i.e. recognising the branches which contain μ-trails, and then determinise and complement it into the required automaton. This is where Safra's determinisation construction can be used. It is from studying the structure of the global automaton obtained from the Safra construction that leads us to the tableau system TS presented in this thesis.

In what follows, we shall look at the global automaton obtained using Safra construction for the formulae in the restricted class $\Sigma_2^\mu(1)$, which contains all the Σ_2^μ-formulae with one μ-variable (i.e., roughly speaking, a formula ϕ is in $\Sigma_2^\mu(1)$ if it has one μ-variable which is an outermost variable in the formula). We then show that the obtained Safra automaton is closely related to the restriction of the tableau system TS where the initial formula is in $\Sigma_2^\mu(1)$. The general version of the tableau system TS is

a straightforward generalisation of such restricted tableau system.

Fix a closed and guarded formula ϕ in $\Sigma_2^\mu(1)$, and suppose $\mu Z.\psi_Z$ is the (only) μ-formula in ϕ.

Basically, the input into the global automaton is an infinite sequence of the goals along a branch in a TS_0-tableau for ϕ (in this case, a goal is a set of subformulae of ϕ). But since the automaton needs to know the trail structure along such branch, we shall include the sequence of the tableau rules applied along the branch as part of the input. Formally, let Σ be the set of all triples $(\Gamma, \mathsf{R}, \Gamma')$, where Γ, Γ' are sets of subformulae of ϕ and R is a tableau rule in TS_0 such that R can be applied on Γ to obtain Γ'. Let $\mathcal{B} \subseteq \Sigma^\omega$ contain all words $(\Gamma_0, \mathsf{R}_0, \Gamma_0')(\Gamma_1, \mathsf{R}_1, \Gamma_1')...$ such that $\Gamma_0 = \{\phi\}$ and $\Gamma_i' = \Gamma_{i+1}$ for each $i \geq 0$. Each branch $u_0u_1u_2...$ in a TS_0-tableau determines a word $(\Gamma_0, \mathsf{R}_0, \Gamma_0')(\Gamma_1, \mathsf{R}_1, \Gamma_1')...$ in $\mathcal{B}$ in the obvious way, i.e. for each $i \geq 0$, Γ_i is the goal at node u_i and R_i is the tableau rule applied at u_i. With some abuse in terminology, a word $b \in \mathcal{B}$ will also be called a "branch".

We can define the notion of trails on a branch $b \in \mathcal{B}$ in the straightforward way. Given $(\Gamma, \mathsf{R}, \Gamma') \in \Sigma$ and formulae $\psi \in \Gamma$, $\psi' \in \Gamma'$, we shall write $(\Gamma, \psi) \xrightarrow{\mathsf{R}} (\Gamma', \psi')$ if, upon applying the rule R on Γ to obtain Γ', either ψ is reduced to ψ' or ψ is not affected by the application of R and $\psi = \psi'$ (this relation is analogous to the notion of *dependency relations* on tableaux; see Definition 3.32). Suppose $b = (\Gamma_0, \mathsf{R}_0, \Gamma_0')(\Gamma_1, \mathsf{R}_1, \Gamma_1')...$ is in $\mathcal{B}$. A *trail* on b is a sequence $\tau = \psi_0, \psi_1, ...$ such that $\psi_i \in \Gamma_i$ and $(\Gamma_i, \psi_i) \xrightarrow{\mathsf{R}_i} (\Gamma_{i+1}, \psi_{i+1})$, for each $i \geq 0$. The usual terminology for trails applies here. In particular, X is *unfolded infinitely often* on τ iff τ contains infinitely many occurrences of the sequence X, ψ_X (where ψ_X is the body of the fixpoint formula identified by X). τ is called a *μ-trail* iff the highest variable unfolded infinitely on τ is a μ-variable (in this case, Z).

As mentioned, we shall start by constructing an automaton recognising every branch which contains a trail in which the μ-variable Z is *unfolded* infinitely often. The automaton works by nondeterministically choosing a trail on the input branch and checking that Z is unfolded infinitely often on the trail. At any time, the automaton remembers one formula and also whether the formula is the unfolding of Z in the previous step. So we let the states of the automaton be the subformulae of ϕ together with a special formula ψ_Z^*. Basically, the superscript $*$ is a flag which indicates that the rule Unfold_μ is applied to Z in the previous step. Precisely, the nondeterministic Büchi automaton $\mathcal{N}_\phi = \langle \Sigma, Q, q_0, \delta, F \rangle$ can be given as follows:

- $Q = \mathrm{Sub}(\phi) \cup \{\psi_Z^*\}$,
- $q_0 = \phi$,
- For each $\gamma \in Q$ and $(\Gamma, \mathsf{R}, \Gamma') \in \Sigma$,
  - (a) if $\gamma = Z$ and $\mathsf{R} = \mathsf{Unfold}_\mu$, then $\delta(\gamma, (\Gamma, \mathsf{R}, \Gamma')) = \{\psi_Z^*\}$; otherwise
  - (b) if $\gamma = \psi_Z^*$ then $\delta(\gamma, (\Gamma, \mathsf{R}, \Gamma')) = \{\gamma' \mid (\Gamma, \psi_Z) \xrightarrow{\mathsf{R}} (\Gamma', \gamma')\}$;
  - (c) if $\gamma \neq \psi_Z^*$ then $\delta(\gamma, (\Gamma, \mathsf{R}, \Gamma')) = \{\gamma' \mid (\Gamma, \gamma) \xrightarrow{\mathsf{R}} (\Gamma', \gamma')\}$,

- $F = \{\psi_Z^*\}$.

It is quite clear that $\mathcal{N}_\phi$ accepts a branch $b \in \mathcal{B}$ iff there is a trail in which Z is unfolded infinitely often on b. Note that, since ϕ is assumed to be guarded, $\mathcal{N}_\phi$ is equivalent to an automaton which detects a trail in which Z *occurs* infinitely often (such an automaton can be obtained by replacing the Büchi condition F above by $\{Z\}$). However, we need the automaton $\mathcal{N}_\phi$ as described above so that the resulting Safra automaton resembles our tableau system TS.

We shall now apply Safra's determinisation construction [Saf88] on $\mathcal{N}_\phi$. The main ingredient of the construction is the *Safra trees*, which can be defined formally as follows. Suppose $|Q| = n$. Fix a set $U = \{u_1, ..., u_n\}$ of *nodes*. A *Safra tree* (for $\mathcal{N}_\phi$) is a tuple $(T, \trianglelefteq, L, c)$ consisting of the following components:

- T is a finite tree whose nodes are drawn from U.
- $\trianglelefteq$ is a partial order relating every pair of nodes with a common parent. Thus, $\trianglelefteq$ linearly orders the children of each node in T. This ordering induces the relation "Left" on the nodes of T: Left(u, v) iff there are two nodes $u' \neq v'$ such that $u' \trianglelefteq v'$ and u is a descendant of u' and v is a descendant of v'.
- $L : T \to \wp(Q)$ is a function assigning a set of states in Q to each node in T ($L(u)$ is called the *label* of node u).
- $c : T \to \{\text{White}, \text{Green}\}$ is a function assigning a *colour*, White or Green, to each node in T.

In addition, a Safra tree must also satisfy the following conditions:

- the label of each node is a *proper* superset of the union of the labels of its children,
- the labels of any two non-ancestral nodes are disjoint.

The *Safra automaton* for $\mathcal{N}_\phi$ is the deterministic Rabin automaton $\mathcal{D}_\phi = \langle \Sigma, Q', \tau_0, \delta', C \rangle$ given as follows:

- Q' contains all Safra trees for $\mathcal{N}_\phi$.
- The initial state τ_0 is the Safra tree with a single node labelled by $\{\phi\}$ and coloured White.
- For each $(\Gamma, \mathsf{R}, \Gamma') \in \Sigma$ and $\tau \in Q'$, $\delta'(\tau, (\Gamma, \mathsf{R}, \Gamma'))$ is the Safra tree obtained by *sequentially* applying the following operations on τ:
  - A1. for each node u in τ, replace the label of u by $\bigcup_{\gamma \in L(u)} \delta(\gamma, (\Gamma, \mathsf{R}, \Gamma'))$;
  - A2. for each node u, if $L(u) \cap F \neq \emptyset$, create a new son v to the right of all existing children of u. Set $L(v) = L(u) \cap F$ (in other words, if $L(u)$ contains ψ_Z^*, then create a new rightmost son v labelled by $\{\psi_Z^*\}$);
  - A3. if both $L(u)$ and $L(v)$ contain γ and Left(u, v), then remove γ from $L(v)$;
  - A4. remove all the nodes with the empty label;
  - A5. for each node u whose label *equals* the union of the labels of its children, remove all nodes below u and set $c(u) =$ Green. Set the colours of other nodes to White.

- C is the Rabin acceptance condition which accepts all the runs where, for some i, u_i is coloured Green infinitely often and u_i is removed finitely often.

The complement automaton $\mathcal{A}_\phi$ can be obtained from $\mathcal{D}_\phi$ by replacing the Rabin acceptance condition C by the *Streett* condition $\overline{C}$ such that $\overline{C}$ accepts a run iff, for each i $(1 \leq i \leq n)$ if u_i is coloured Green infinitely often, then it is removed infinitely often.

From the above construction, it is not difficult to see that a goal in a TS-tableau has a similar structure as a Safra tree. Suppose $\Theta \vdash \Gamma$ is a goal at some node u in a TS-tableau for ϕ. Let $U = \{\epsilon, z^1, ..., z^{|\phi|}\}$ where each z^i is a name for Z and ϵ is any symbol distinct from $z^1, ..., z^{|\phi|}$. Define the corresponding Safra tree $\tau = (T, \trianglelefteq, L, c)$ as follows:

- The nodes T include ϵ and all the names appearing in the goal. ϵ is the root of T and, for any names z^i, z^j in T, z^j is a child of z^i iff there is a name sequence of the form $\rho \cdot z^i \cdot z^j \cdot \rho'$ in Γ.
- $\trianglelefteq$ relates every pair of (non-root) nodes z^i, z^j with a common parent: $z^i \trianglelefteq z^j$ iff either $z^i = z^j$ or z^i occurs before z^j in Θ.
- $L(\epsilon) = \{\psi \,|\, \exists \rho.\psi^\rho \in \Gamma\}$, and for each name z^i in T, $L(z^i) = \{\psi \,|\, \exists \rho.\psi^\rho \in \Gamma$ and z^i is in $\rho\}$.
- For each name z^i, $c(z^i) =$ Green if the rule Reset_{z^i} is applied in the parent node; otherwise $c(z^i) =$ White.

From this representation of goals as Safra trees, the tableau rules in TS can be seen as the counterparts of the operations A1 - A5 above. This is particularly clear if we consider the following simplifications of the rules Unfold_σ, Reset_z, and Thin for the case where the initial formula is in $\Sigma_2^\mu(1)$:

$$\mathsf{Unfold}'_\mu: \quad \frac{\Theta \vdash Z^\rho, \Gamma}{\Theta' \cdot z^i \vdash \psi_Z^{\rho \cdot z^i}, \Gamma} \qquad \begin{array}{l} Z \text{ identifies } \mu Z.\psi_Z \text{ and} \\ z^i \text{ is the } \textit{first} \text{ name for } Z \textit{ not} \text{ occurring in } \Theta. \end{array}$$

$$\mathsf{Unfold}'_\nu: \quad \frac{\Theta \vdash X^\rho, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma} \quad X \text{ identifies } \nu X.\psi.$$

$$\mathsf{Reset}'_z: \quad \frac{\Theta \vdash \psi_1^{\rho \cdot z \cdot z_1 \cdot \rho_1}, ..., \psi_n^{\rho \cdot z \cdot z_n \cdot \rho_n}, \Gamma}{\Theta' \vdash \psi_1^{\rho \cdot z}, ..., \psi_n^{\rho \cdot z}, \Gamma} \quad n \geq 1$$

where z does *not* occur in Γ.

$$\mathsf{Thin}': \quad \frac{\Theta \vdash \psi^\rho, \psi^{\rho'}, \Gamma}{\Theta' \vdash \psi^\rho, \Gamma} \quad \text{if } \rho \prec_\Theta \rho' \text{ or } \rho' \text{ is a } \textit{proper} \text{ prefix of } \rho.$$

Note that in our case (where there is only one μ-variable), $\rho \prec_\Theta \rho'$ iff, for some j, $\rho(j)$ occurs before $\rho'(j)$ in Θ and $\rho(i) = \rho'(i)$ for each $i < j$. Notice how the manipulations

of name sequences in the rules Unfold'_μ, Thin', and Reset'_z resemble the operations A2, A3, and A5, respectively.

Once we have learned what the tableau rules should be for the case where $\phi \in \Sigma_2^\mu(1)$, it is straightforward to generalise them for the full logic. This is in fact how we obtain the tableau system TS in the first place. It is also interesting to note that one should be able to obtain the tableau system for the full logic by first extending the automaton $\mathcal{N}_\phi$ so that it also works for any formula ϕ outside $\Sigma_2^\mu(1)$. A straightforward extension would employ the *parity* acceptance condition instead of the Büchi one. From automata theory, such a parity automaton can be converted into an equivalent Büchi automaton. We can then apply the Safra construction as above to this equivalent Büchi automaton. We did not choose this route as the obtained Safra automaton is quite complicated. However, it would be interesting to see how this latter Safra automaton relates to the tableau system TS.

4.4 Axiomatic Completeness

One of our goals of research is to prove the completeness of the axiom system **AX** (Definition 2.32). This is one of the reasons we study tableaux for the modal μ-calculus. It is hoped that the sound and complete tableau system we have obtained can be used to show the completeness of **AX** in the same way tableaux have been used for this purpose for many other logics. So far, we have only obtained a completeness proof for the aconjunctive fragment based on the tableau system ACON described earlier. The completeness for this sublogic has already been shown by Kozen [Koz83], also by a tableau method. But we believe our proof based on ACON is more transparent and easier to comprehend.

4.4.1 Canonical Models

Before we proceed, let us briefly mention the method of proving the completeness of axiomatisation using *canonical models* ([BdV01],[Gol92]). This traditional method fails to give the completeness for the modal μ-calculus. Essentially, it is the construction of a model from the formulae in the logic in the same way as in Henkin's completeness proof of the first-order logic. Let us try using this technique for the modal μ-calculus.

The canonical model $\mathcal{M}_\Lambda$ for an axiom system Λ is built up from maximally Λ-consistent sets of formulae. For many axiom systems, including **AX**, it can be shown that every consistent set can be extended to a maximal one (this is usually called a *Lindenbaum lemma*). The canonical model $\mathcal{M}_\Lambda$ is $\langle S, \{R_a\}_{a\in \mathrm{Act}}, \mathcal{V}_{\mathrm{Prop}}\rangle$ where

- S is the set of all maximally Λ-consistent sets of formulae;
- $\Gamma R_a \Gamma'$ iff, for all ϕ, $[a]\phi \in \Gamma$ implies $\phi \in \Gamma'$;
- $\mathcal{V}_{\mathrm{Prop}}(P) = \{\Gamma \in S \mid P \in \Gamma\}$.

For many systems of modal logic, including the system **K**, the canonical model defined above has the property that

for each formula ϕ, $\mathcal{M}_\Lambda, \Gamma \models \phi$ iff $\phi \in \Gamma$.

With this property, a (finite or infinite) set Γ is satisfiable in $\mathcal{M}_\Lambda$ iff Γ is Λ-consistent. This not only implies the completeness of Λ but also shows that the logic is compact, i.e. every unsatisfiable set of formulae contains a finite unsatisfiable subset. But it is known that the modal μ-calculus does not have the compactness property. To see this, consider the set Φ

$$\mu Z.\neg P \vee \langle a \rangle Z,\ P,\ [a]P,\ [a][a]P,\ [a][a][a]P,\ ...$$

Clearly, every finite subset of Φ is satisfiable, but not Φ itself. This implies that **AX** does not have the above property.

This problem also arises when we apply the technique for the temporal logic LTL, CTL, or PDL. For these logics, the canonical model $\mathcal{M}_\Lambda$ can be transformed so that it has the required property. Roughly, the model is first *filtrated* through a *finite* subformula-closed set Γ to obtain a finite model $\mathcal{M}^\Gamma_\Lambda$, and then the model is *carefully* unravelled into a new model $\mathcal{M}'$ having the property that, for each $\phi \in \Gamma$, ϕ is satisfiable in $\mathcal{M}'$ iff ϕ is Λ-consistent. Notice that this does *not* imply compactness as it is required that Γ is finite. For the modal μ-calculus, the difficulty is how to unravel $\mathcal{M}^\Gamma_{\mathbf{AX}}$ into the model $\mathcal{M}'$ satisfying the mentioned property. In a sense, this is basically the reason why it is difficult to obtain a finitistic tableau system for the modal μ-calculus. It is hoped that a better understanding of the tableau system TS may lead to a useful technique for manipulating the canonical model.

4.4.2 Completeness for Aconjunctive Fragment

There is a simple proof of the completeness of axiom system **AX** with respect to the aconjunctive fragment which utilises the tableau system ACON. It is however more convenient if we proceed the proof using a slight variant of ACON. Namely, the tableau system ACON' has the same tableau rules as ACON but with stronger conditions for termination and success. We first explain when a leaf is counted as a successful terminal or an unsuccessful terminal. A branch in a tableau terminates once a terminal of either type is reached.

A *successful terminal* in an ACON'-tableau is a leaf u such that *one* of the following holds:

S1. The goal at u contains only literals and $[\cdot]$-formulae, but not any complementary pair of literals.

S2. There is a node v above u with the same goal such that, for each name z, if a μ-variable Z^ρ, where $\rho(Z) = z$, is unfolded between v and u, then there is a goal on the path from v to u where z does *not* occur.

An *unsuccessful terminal* in an ACON'-tableau is a leaf u which is *not* a successful terminal and, additionally, satisfies *one* of the following:

U1. The goal at u contains a complementary pair of literals.

U2. There is a path from some node v above u of the form:

$$v : \Theta \vdash Z^{\rho}, \Gamma$$
$$\vdots$$
$$u : \Theta \vdash Z^{\rho}, \Gamma$$

such that

- Z^{ρ} is unfolded at v,
- the name $\rho(Z)$ occurs in every goal on the path, and
- no μ-variable $Y^{\rho'}$, where $\rho'(Y) <_{\Theta} \rho(Z)$, unfolded on the path.

A *successful tableau* is a finite tableau all whose leaves are successful terminals.

It can be shown that every tableau in ACON' must be finite. Note that, as in Lemma 4.10, the names for each μ-variable Z used in an ACON'-tableau are among $z^1, ..., z^{|\phi|}$. Hence the number of possible goals in a tableau is finite.

Lemma 4.57. *Every tableau in ACON' is finite.*

Proof. Suppose $\mathcal{T}$ is an infinite tableau in ACON'. $\mathcal{T}$ must contain an infinite branch $u_0, u_1, ...$ such that, for each prefix $u_0, ..., u_n$ of this branch, u_n is neither a successful terminal nor an unsuccessful terminal. Since there are finitely many possible goals, there must be some node u_m such that, for each node u_i, $i \geq m$, the goal at u_i occurs infinitely often on the branch. Consider the path $u_m, u_{m+1}, ...$ There must be a sequence of nodes (not necessarily consecutive)

$$v_1 : \Theta \vdash \Gamma, \quad v_2 : \Theta \vdash \Gamma, \quad ...$$

along this path such that all v_i have the same goal $\Theta \vdash \Gamma$. For each $i \geq 2$, since v_i is not a successful terminal, there must be a name z_i (for some μ-variable Z_i) in Θ such that z_i occurs in every goal on the path from v_1 and v_i, and some formula $Z_i^{\rho_i}$ where $\rho_i(Z_i) = z_i$ is unfolded at some node on the path. Consider the names $z_2, z_3, ...$ Suppose z is the least name with respect to Θ in $\{z_2, z_3, ...\}$ (i.e. z occurs in Θ before (or at the same position as) each z_i). It is easily seen that z must occur in every goal on the branch from node v_1 onwards. The reason is that if z does not occur in the goal at some node between v_1 and some v_i, then $z \neq z_i$ and z cannot occur before z_i in Θ (because z_i occurs at every goal from v_1 to v_i, and when z is added to the global sequence at some node, it is added at the end of the global sequence and hence must appear after z_i in Θ).

Since there are finitely many possible goals, what we have just shown implies that there is a name z in Θ such that

- z occurs in every goal on the branch from node v_1 onwards, and
- a μ-variable Z^ρ, where $\rho(Z) = z$, is unfolded at some node v on the branch below v_1.

Let z be the first name in Θ with this property. Since every goal occurring at some node after v_1 occurs infinitely often on the branch, there must be a path $v : \Theta' \vdash Z^\rho, \Gamma'$, ..., $u : \Theta' \vdash Z^\rho, \Gamma'$ such that $\rho(Z) = z$ occurs throughout the path and Z^ρ is unfolded at v. We claim that there is no μ-variable $Y^{\rho'}$, where $\rho'(Y) <_{\Theta'} z$, unfolded on this path. If this is not the case (hence $\rho'(Y)$ occurs in Θ' before z), since z occurs in every goal on the branch below node v_1, the name $\rho'(Y)$ must occur before z in every global sequence along the branch from v_1 onwards. But we would have chosen the name $\rho'(Y)$ instead of z. This implies that u is an unsuccessful terminal, contradicting the fact that every node on an infinite branch is neither a successful terminal nor an unsuccessful terminal. Hence ϕ cannot have an infinite tableau. □

The soundness of this tableau system can be shown in exactly the same way as for ACON.

Theorem 4.58. *Every closed and guarded formula which has a successful ACON'-tableau is satisfiable.*

Proof. Same as Theorem 4.20. □

We now turn to prove a stronger version of the completeness of ACON', namely that every ***AX**-consistent* formula has a successful tableau in ACON'. This implies that the axiom system **AX** is complete for the aconjunctive fragment.

The completeness proof relies on the following important property, which was first employed in [Koz83], and more recently in completeness proofs of LTL and CTL in [LS01].

Proposition 4.59. *If $\{\mu Z.\psi(Z)\} \cup \Gamma$, where Z does* not *occur free in Γ, is consistent in **AX**, then so is $\{\psi(\mu Z.(\neg \bigwedge \Gamma) \wedge \psi(Z))\} \cup \Gamma$.*

Proof. This follows from Proposition 2.38(f). □

Strengthening. In the proof below, we adopt a shorthand notation from [LS01], called a *strengthening* of a fixpoint formula $\mu Z.\psi$:

$$\mu Z.\psi_{\neg\gamma_1,...,\neg\gamma_n} \quad = \quad \mu Z.\neg\gamma_1 \wedge ... \wedge \neg\gamma_n \wedge \psi.$$

Conversely, the *unstrengthened version* of $\mu Z.\psi_{\neg\gamma_1,...,\neg\gamma_n}$ is the original formula $\mu Z.\psi$ identified by Z. Note that only the strengthening of μ-formulae are used here.

Theorem 4.60. *Every consistent aconjunctive (closed) formula has a successful ACON'-tableau.*

Proof. Suppose ϕ is a consistent aconjunctive formula, and let $X_1, ..., X_n$ be all the variables in ϕ such that X_i higher than X_j implies $i < j$. We will construct a successful ACON'-tableau for ϕ. To guide the construction, we associate a fixpoint formula to each name in a goal. This is similar to the use of *definition lists* in [SW91]. Particularly, in the construction below, the *global sequence* Θ in a goal will be a sequence of *definitions*:

$$\Theta \quad = \quad \langle z_1 = \psi_1, ..., z_n = \psi_n \rangle, \qquad (n \geq 0)$$

where $z_1, ..., z_n$ are distinct names. Moreover, if z_i is a name for μ-variable Z_i, ψ_i will be a strengthening of the fixpoint formula $\mu Z_i.\psi$ identified by Z_i. If Θ is in the above form, we let the *domain* of Θ be $\{z_1, ..., z_n\}$ and, for each i, $\Theta(z_i) = \psi_i$.

Definition 4.61. Suppose Θ is a global sequence. For each augmented formula ψ^ρ where all the names in ρ are in Θ, define[6]

$$\psi^\rho \cdot \Theta \quad = \quad \psi\{\gamma_n/X_n\}...\{\gamma_1/X_1\},$$

where, for each i, $1 \leq i \leq n$,

- if X_i is a μ-variable and there is a name z for X_i in ρ, then $\gamma_i = \Theta(z)$;
- otherwise $\gamma_i = \sigma X_i.\psi_i$, the fixpoint formula identified by X_i.

More generally, $\Gamma \cdot \Theta = \bigwedge\{\psi^\rho \cdot \Theta \mid \psi^\rho \in \Gamma\}$, for any set Γ.

Definition 4.62. A goal $\Theta \vdash \Gamma$ is said to be *consistent* iff $\Gamma \cdot \Theta$ is consistent in **AX**.

The following properties are used in the construction. Note that the least-fixpoint counterpart of (f) will be explained later in the construction.

Lemma 4.63. *Below Θ' denotes the result of removing all the definitions for the names not occurring in the goal on the right hand side.*

(a) If $\Theta \vdash \psi^\rho, \psi'^{\rho'}, \Gamma$ is consistent, then so is $\Theta' \vdash \psi^\rho, \Gamma$.
(b) If $\Theta \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma$ is consistent, then so is $\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma$.
(c) If $\Theta \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma$ is consistent, then so is $\Theta \vdash \psi_i^\rho, \Gamma$, for some $i \in \{1, 2\}$.
(d) If $\Theta \vdash (\mu Z.\psi)^\rho, \Gamma$ is consistent, then so is $\Theta \cdot \langle z = \mu Z.\psi \rangle \vdash Z^{\rho \cdot z}, \Gamma$ where z is any name for Z not occurring in Θ.
(e) If $\Theta \vdash (\nu X.\psi)^\rho, \Gamma$ is consistent, then so is $\Theta \vdash X^\rho, \Gamma$.
(f) If $\Theta \vdash X^\rho, \Gamma$, where X identifies $\nu X.\psi$, is consistent, then so is $\Theta' \vdash \psi^{\rho \restriction X}, \Gamma$.
(g) If $\Theta \vdash (\langle a \rangle \psi)^\rho, \Gamma$ is consistent, then so is $\Theta' \vdash \psi^{\rho_i}, \Gamma_a$ where $\Gamma_a = \{\psi^\rho \mid ([a]\psi)^\rho \in \Gamma\}$.

Proof. These follows quite easily from the basic properties provable in the axiom system **AX**. We explain part (f) only.

[6] As usual, $\psi\{\gamma_2/X_2\}\{\gamma_1/X_1\}$ means $(\psi\{\gamma_2/X_2\})\{\gamma_1/X_1\}$.

(f) Suppose $\Theta \vdash X^\rho, \Gamma$, where X identifies $\nu X.\psi$, is consistent; hence by definition $(\{X^\rho\} \cup \Gamma) \cdot \Theta$ is consistent. From Definition 4.61, if $X = X_i$ then

$$\begin{aligned} X^\rho \cdot \Theta &= X\{\gamma_n/X_n\}...\{\nu X.\psi/X_i\}...\{\gamma_1/X_1\} \\ &= \nu X.\psi\{\gamma_{i-1}/X_{i-1}\}...\{\gamma_1/X_1\}. \end{aligned}$$

Since $\vdash \nu X.\psi \leftrightarrow \psi\{\nu X.\psi/X\}$, applying Proposition 2.35, we can deduce that

$$\vdash X^\rho \cdot \Theta \quad \leftrightarrow \quad \psi\{\nu X.\psi/X_i\}\{\gamma_{i-1}/X_{i-1}\}...\{\gamma_1/X_1\}.$$

But the latter formula equals to $\psi^{\rho \restriction X} \cdot \Theta$. It follows that that the goal $\Theta' \vdash \psi^{\rho \restriction X}, \Gamma$ is consistent. □

Lemma 4.64. *Suppose Θ and Θ' are global sequences with the same domain. If $\vdash \Theta(z) \rightarrow \Theta'(z)$ for each name z in the domain, then $\vdash \psi^\rho \cdot \Theta \rightarrow \psi^\rho \cdot \Theta'$ for any augmented formula ψ^ρ where all the names in ρ are defined in Θ.*

Proof. Suppose $\vdash \Theta(z) \rightarrow \Theta'(z)$ for each name z in the domain. From Definition 4.61,

$$\begin{aligned} \psi^\rho \cdot \Theta &= \psi\{\gamma_n/X_n\}...\{\gamma_1/X_1\}, \\ \psi^\rho \cdot \Theta' &= \psi\{\gamma'_n/X_n\}...\{\gamma'_1/X_1\}, \end{aligned}$$

where, for each i, $\gamma_i = \Theta(z)$ and $\gamma'_i = \Theta'(z)$ if X_i is a μ-variable and there is a name z for X_i in the domain, otherwise $\gamma_i = \gamma'_i$ is the fixpoint formula identified by X_i. By monotonicity (Proposition 2.37), we conclude that $\vdash \psi^\rho \cdot \Theta \rightarrow \psi^\rho \cdot \Theta'$. □

Construction. Starting with the smallest tableau $\mathcal{T}_0$ for ϕ, we subsequently expand the tableau while preserving the consistency of each goal (the initial goal is consistent by assumption). Suppose we have so far constructed $\mathcal{T}_0, ..., \mathcal{T}_i$, all whose goals are consistent. We expand each non-terminal leaf $\Theta \vdash \Gamma$ in $\mathcal{T}_i$ following one of the cases below, giving higher priority to the earlier ones:

- $\Gamma = \psi^\rho, \psi^{\rho'}, \Gamma'$. Apply Thin to create a subgoal $\Theta' \vdash \psi^\rho, \Gamma'$ or $\Theta' \vdash \psi^{\rho'}, \Gamma'$. By Lemma 4.63(a), both of these subgoals are consistent.
- $\Gamma = (\psi_1 \wedge \psi_2)^\rho, \Gamma'$. Apply $\mathsf{R}\wedge$ to create subgoal $\Theta \vdash \psi_1^\rho, \psi_2^\rho, \Gamma'$. By Lemma 4.63(b), the subgoal is consistent.
- $\Gamma = (\psi_1 \vee \psi_2)^\rho, \Gamma'$. Since the goal is consistent, by Lemma 4.63(c), there must be some i such that the subgoal $\Theta \vdash \psi_i^\rho, \Gamma'$ is consistent. Apply rule $\mathsf{R}\vee$ to create such subgoal.
- $\Gamma = \mu Z.\psi^\rho, \Gamma'$. Apply $\mathsf{R}\mu$ to create subgoal $\Theta \cdot \langle z = \mu Z.\psi \rangle \vdash Z^{\rho \cdot z}, \Gamma'$, where z is a name for Z not occurring in Θ. By Lemma 4.63(d), the subgoal is consistent.
- $\Gamma = \nu X.\psi^\rho, \Gamma'$. Apply $\mathsf{R}\nu$ to create subgoal $\Theta \vdash X^\rho, \Gamma'$. By Lemma 4.63(e), the subgoal is consistent.

- $\Gamma = Z^\rho, \Gamma'$. Apply Unfold_μ to create subgoal $\Theta'' \vdash \psi^{\rho\restriction Z}, \Gamma'$, where Θ'' is given as follows. Suppose $\rho(Z) = z$. First, obtain Θ' by updating the definitions as follows:
  - $\Theta'(y) = \Theta(y)$, for each $y <_\Theta z$,
  - $\Theta'(y)$ is the *unstrengthened* version of $\Theta(y)$, for each $y \geq_\Theta z$.

  Obviously $\vdash \Theta(y) \to \Theta'(y)$ for each name y in the domain. Since $(\{Z^\rho\} \cup \Gamma') \cdot \Theta$ is consistent, by Lemma 4.64, so is $(\{Z^\rho\} \cup \Gamma') \cdot \Theta'$.
  Next, obtain Θ'' by removing all the definitions in Θ' for the names not occurring in $\psi^{\rho\restriction Z}, \Gamma'$ and updating the definition for z as follows:
  - $\Theta''(z) = \mu Z.\psi_{\neg\gamma_1,\ldots,\neg\gamma_n,\neg\gamma}$ if $\Theta(z) = \mu Z.\psi_{\neg\gamma_1,\ldots,\neg\gamma_n}$ and $\gamma = \Gamma' \cdot \Theta'$.

  By Lemma 4.9, there is no augmented formula $\psi'^{\rho'}$ in Γ' where Z is active in ψ' and $\rho'(Z) = z$. This implies that $\Gamma' \cdot \Theta' = \Gamma' \cdot \Theta''$. By Proposition 4.59, since $(\{Z^\rho\} \cup \Gamma') \cdot \Theta'$ is consistent, so is

  $$\{\psi^{\rho\restriction Z} \cdot \Theta''\} \cup (\Gamma' \cdot \Theta')$$

  and hence so is $(\{\psi^{\rho\restriction Z}\} \cup \Gamma') \cdot \Theta''$, which implies that the subgoal is consistent.
- $\Gamma = X^\rho, \Gamma'$. Apply Unfold_ν to create subgoal $\Theta' \vdash \psi^{\rho\restriction X}, \Gamma'$. By Lemma 4.63(f), the subgoal is consistent.
- $\Gamma = (\langle a_1\rangle\psi_1)^{\rho_1}, \ldots, (\langle a_n\rangle\psi_n)^{\rho_n}, \Gamma'$ where $n \geq 1$ and Γ' contains only literals and $[\cdot]$-formulae. Apply $\mathsf{R}\langle\rangle$ to create n subgoals $\Theta_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}$ $(1 \leq i \leq n)$. By Lemma 4.63(g), each of these subgoals is consistent.

By Lemma 4.57, the construction must terminate at some tableau $\mathcal{T}'$ all whose leaves are terminal. Clearly since each goal in $\mathcal{T}'$ is consistent, all the leaves which contain only literals and/or $[\cdot]$-formulae are successful. Other leaves in $\mathcal{T}'$ must also be successful. Assume otherwise. Thus there is a path to an unsuccessful terminal u_n:

$$u_1 : \Theta_1 \vdash Z^\rho, \Gamma_1, \quad \ldots, \quad u_n : \Theta_n \vdash Z^\rho, \Gamma_n$$

such that

- $\Gamma_1 = \Gamma_n$ and $\hat{\Theta}_1 = \hat{\Theta}_n$ (where $\hat{\Theta}$ is the sequence of names obtained from removing all the defining formulae in Θ),
- Z^ρ is unfolded at u_1,
- the name $\rho(Z)$ occurs in each Θ_i,
- no μ-variable $Y^{\rho'}$, where $\rho'(Y) <_\Theta \rho(Z)$, unfolded on the path.

Suppose $\rho(Z) = z$. From the construction, the last two conditions clearly imply that the prefix of each Θ_i up to, but *not* including, the definition for z are the same. Thus $\Theta_1(y) = \Theta_n(y)$ for each name $y <_{\Theta_1} z$. Since Z^ρ is unfolded at u_1, from the above

construction, $\Theta_2(z)$ must be of the form

$$\Theta_2(z) \quad = \quad \mu Z.\rho_{\neg\gamma_1,\ldots,\neg(\Gamma_1\cdot\Theta_1')},$$

where, for each name $y <_{\Theta_1} z$, $\Theta_1'(y) = \Theta_1(y)$ and, for each name $y \geq_{\Theta_1} z$, $\Theta_1'(y)$ is the *unstrengthened* version of $\Theta_1(y)$. Thus $\Theta_n(z)$ must be of the form

$$\Theta_n(z) \quad = \quad \mu Z.\rho_{\neg\gamma_1,\ldots,\neg(\Gamma_1\cdot\Theta_1'),\ldots,\neg\gamma_m}.$$

This implies that $\vdash \Theta_n(y) \rightarrow \Theta_1'(y)$ for each name y in the domain. By Lemma 4.64 (and the fact that $\Gamma_n = \Gamma_1$), it follows that

$$\vdash \Gamma_n \cdot \Theta_n \rightarrow \Gamma_1 \cdot \Theta_1'.$$

But since

$$\vdash Z^\rho \cdot \Theta_n \rightarrow \neg(\Gamma_1 \cdot \Theta_1'),$$

this implies that $(\{Z^\rho\} \cup \Gamma_n) \cdot \Theta_n$ is inconsistent, contradicting the fact that the goal $\Theta_n \vdash Z^\rho, \Gamma_n$ is consistent. Hence every terminal in $\mathcal{T}'$ is successful. By replacing the global sequence Θ in each goal in $\mathcal{T}'$ by $\hat{\Theta}$, we obtain a successful $\mathsf{ACON'}$-tableau for ϕ. □

Remark 4.65. Roughly speaking, the idea of the previous proof is that whenever a μ-variable Z in a goal $\Theta \vdash Z^\rho, \Gamma$ is unfolded, the μ-formula $\mu Z.\psi$ associated with the name $\rho(Z)$ in Θ is *strengthened* to $\mu Z.\psi_{\neg\Gamma\cdot\Theta'}$, where Θ' is the result of *resetting* each μ-formula for the names appearing later than $\rho(Z)$ in Θ to original. By Proposition 4.59, the formula $\psi\{\mu Z.\psi_{\neg\Gamma\cdot\Theta'}/Z\} \wedge \Gamma \cdot \Theta'$ is consistent. This strategy guarantees that an unsuccessful terminal is never reached.

This technique might not work if the initial formula ϕ is non-aconjunctive. The reason is, in a goal $\Theta \vdash Z^\rho, \Gamma$, there might be a formula $\gamma^{\rho'}$ in Γ where Z is active and $\rho'(Z) = \rho(Z)$. Obviously we cannot associate the strengthened formula $\mu Z.\psi_{\Gamma\cdot\Theta'}$ to the variable Z in $\gamma^{\rho'}$ (see Proposition 4.59). In other words, after the unfolding, Z in the unfolding ψ^ρ of Z^ρ and Z in $\gamma^{\rho'}$ should be assigned different formulae. Trying to fix this by introducing a new name for Z in the unfolding ψ^ρ is problematic because it is unclear whether the new name should be ordered before or after $\rho(Z)$ in the global sequence. In fact, it is quite clear that, for non-aconjunctive formulae, associating one name to each μ-variable is insufficient. A more structured method of recording names, such as in tableau system TS, is required. It is then suggestive to try to adapt the above technique to tableau system TS in order to prove the completeness for the full logic. Unfortunately, we are still unable to obtain the proof in this way.

The completeness of the axiom system **AX** for the aconjunctive fragment follows immediately from the previous theorem.

Theorem 4.66 ([Koz83])**.** *Every consistent aconjunctive formula is satisfiable.*

Proof. Suppose there is an aconjunctive formula ϕ which is consistent but not satisfiable. If ϕ is not closed, we simply rename each free variable in ϕ to a distinct proposition letter not occurring in ϕ. Clearly, the renamed formula is consistent (or satisfiable) iff the original formula ϕ is so.

By Proposition 2.41, ϕ is provably equivalent to a guarded formula ϕ'. It is clear that the guarded formula ϕ' obtained from the proof of Proposition 2.41 is still aconjunctive. By the previous theorem, ϕ' has a successful ACON'-tableau. By the soundness of ACON', ϕ' must be satisfiable, contradicting our assumption. Hence every consistent aconjunctive formula must be satisfiable. □

Chapter 5

Model Surgery

As mentioned, the modal μ-calculus has the small model property. This was shown as a consequence of the finiteness result of the tableau system TS (Theorem 4.54 in the previous chapter). As far as we know, apart from our proof based on tableaux, all the known proofs employ automata theory in some form. It is one of our research problems to find a direct proof of the small model theorem. The goal is to find model-theoretic techniques which transform a model of the given formula into a small model. This is analogous to the filtration method in modal logic, where all the states in the model satisfying the same subformulae of the given formula are collapsed into one state. Since the filtration method does not work in the modal μ-calculus, we turn to a more refined approach.

In Chapter 3, the Fundamental Semantic Theorem and its converse establish the connection between models and well-founded pre-models. In particular, we know that a model of the formula can be equipped with a dependency relation to form a well-founded pre-model and, conversely, a well-founded pre-model in which the given formula annotates some state is a model for the formula. Using this connection, to obtain a small model, we start with a well-founded pre-model in which the given formula annotates some state, and then transform it into a small well-founded pre-model. This has led us to study operations on pre-models which preserve well-foundedness. In particular, we define the notion of *trail-equivalence* and study some operations based on this notion. Using this method, we are able to show that every *linear model* can be turned into a small model. But we are still not able to extend this result for arbitrary models. However, for the fragment of the Π^{μ}_{2}-formulae, the small model property can be shown in this way. Additionally, guided by the proof of the small model theorem, we obtain a sound and complete tableau system for the latter sublogic.

Convention. For the rest of this chapter, all formulae are assumed to be in positive normal form.

5.1 Operations on Models and Pre-Models

Basic operations on models, including *disjoint unions* (Definition 2.19) and *generated submodels* (Definition 2.21), are defined in Chapter 3. Such operations can be defined for pre-models in a straightforward way.

Definition 5.1. Let $\mathcal{P}^i = \langle \mathcal{S}^i, \Delta^i, \rightarrow^i \rangle$, $i \in I$, be a family of disjoint pre-models. The *disjoint union* of $\mathcal{P}^i$ ($i \in I$), denoted $\biguplus_{i \in I} \mathcal{P}^i$, is $\langle \biguplus_{i \in I} \mathcal{S}^i, \bigcup_{i \in I} \Delta^i, \bigcup_{i \in I} \rightarrow^i \rangle$.

Definition 5.2. Let $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ be a pre-model. The *pre-model generated by state s in $\mathcal{P}$*, denoted $\mathrm{Sub}_s(\mathcal{P})$, is $\langle \mathrm{Sub}_s(\mathcal{S}), \Delta', \rightarrow' \rangle$, where $\mathrm{Sub}_s(\mathcal{S})$ is the subsystem of $\mathcal{S}$ generated by s, and Δ' and $\rightarrow'$ are the restrictions of Δ and $\rightarrow$ to states in $\mathrm{Sub}_s(\mathcal{S})$, respectively.

It is obvious that the disjoint union of a family of pre-models and the pre-model generated from a state are indeed pre-models (i.e. satisfying the local-consistency conditions, etc.). It is also obvious that these operations preserve well-foundedness.

Proposition 5.3. *For any* well-founded *pre-model $\mathcal{P}$ and disjoint pre-models $\mathcal{P}^i$, $i \in I$, the disjoint union $\biguplus_{i \in I} \mathcal{P}^i$ and the pre-model $\mathrm{Sub}_s(\mathcal{P})$ generated from any state s are well-founded.*

Proof. Obvious. □

In addition to these basic operations, we will be using an operation called *jumping*.

Definition 5.4. Let $\mathcal{S} = \langle S, \{R_a\}_{a \in \mathrm{Act}} \rangle$ be a transition system. A *jump* on $\mathcal{S}$ is any function $f : S \rightarrow S$. Given a jump f, we use $\mathrm{Jump}_f(\mathcal{S})$ to denote $\langle S', \{R'_a\}_{a \in \mathrm{Act}} \rangle$ where

- $S' = S$;
- $R'_a = \{(s, f(t)) \mid sR_at\}$.

For conciseness, we write $\mathrm{Jump}_{s,t}(\mathcal{S})$ for $\mathrm{Jump}_f(\mathcal{S})$, where $f(s) = t$ and $f(s') = s'$ for each state $s' \neq s$. Roughly speaking, in $\mathrm{Jump}_{s,t}(\mathcal{S})$, all the transitions to state s in $\mathcal{S}$ go to t instead.

It is interesting to look at the jumping operation on tree transition systems. The operation $\mathrm{Jump}_{s,t}$ on a tree transition system can have different effects depending on the positions of s and t. For example, suppose $\mathcal{S}$ is the tree transition system shown in Figure 5.1. $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_1,s_4}(\mathcal{S}))$, where s_4 is below s_1, can be seen as the result of replacing the subtree rooted at s_1 by the one rooted at s_4. $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_4,s_1}(\mathcal{S}))$ is resulted from adding a backedge from s_3 to s_1. $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{t_3,s_2}(\mathcal{S}))$, where t_3 and s_2 are on different branches, is the DAG-shaped transition system shown.

This jumping operation will be used extensively in model surgery. Typically, a jump is applied to an annotated structure or a pre-model. In such case, a jump $f : S \rightarrow S$ is required to *respect* the annotation, i.e. the annotation of a state s is the same as that

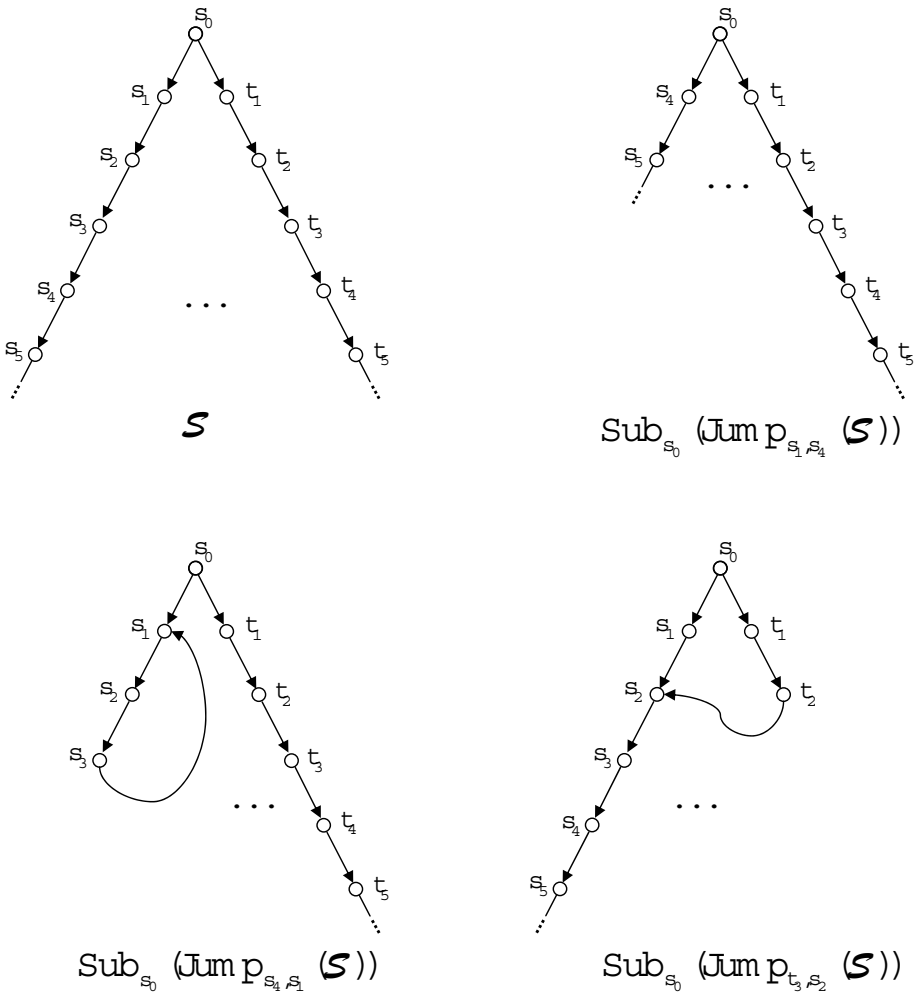


Figure 5.1: Jumping on a tree transition system

of $f(s)$. For example, if we are considering an annotated structure $\langle \mathcal{S}, \Delta \rangle$, only a jump f where $\Delta(s) = \Delta(f(s))$, for each state s, is allowed.

Definition 5.5. Given a pre-model $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ and a jump f which *respects* Δ, $\mathrm{Jump}_f(\mathcal{P})$ denotes $\langle \mathrm{Jump}_f(\mathcal{S}), \Delta, \rightarrow' \rangle$ where $\rightarrow'$ is the smallest relation such that

- $(s, \langle a \rangle \psi) \rightarrow (t, \psi)$ implies $(s, \langle a \rangle \psi) \rightarrow' (f(t), \psi)$,
- $(s, [a]\psi) \rightarrow (t, \psi)$ implies $(s, [a]\psi) \rightarrow' (f(t), \psi)$,
- $(s, \psi) \rightarrow (s, \psi')$ implies $(s, \psi) \rightarrow' (s, \psi')$, if ψ is *not* a modal formula.

It is easy to check that $\mathrm{Jump}_f(\mathcal{P}) = \langle \mathrm{Jump}_f(\mathcal{S}), \Delta, \rightarrow' \rangle$ defined above is indeed a pre-model, i.e. Δ is a locally-consistent annotation on $\mathrm{Jump}_f(\mathcal{S})$ and $\rightarrow'$ is a dependency relation on $\langle \mathrm{Jump}_f(\mathcal{S}), \Delta \rangle$. However, if $\mathcal{P}$ is a well-founded pre-model, $\mathrm{Jump}_f(\mathcal{P})$ is *not* necessarily well-founded. In the rest of this section, we study various situations in which jumping preserves well-foundedness.

5.2 Model Surgery Using Trails

Fix a closed formula ϕ in positive normal form. For the rest of this section, by a pre-model, we mean a pre-model whose states are annotated by subformulae of ϕ. In this section, we study some relations on the states of pre-models for ϕ. The aim is to find an operation based on such relations which can be used to construct a small model from a well-founded pre-model for ϕ.

Notations for trails. For brevity, we introduce some shorthand notation for specifying the existence of certain trails. Suppose $\mathcal{P}$ is a pre-model. For any states s, s', formulae ψ, ψ', and variable X, we write

$$(s, \psi) \stackrel{X}{\Longrightarrow} (s', \psi') \text{ in } \mathcal{P}$$

when there is a trail in $\mathcal{P}$ from (s, ψ) to (s', ψ') in which X is *active*;

$$(s, \psi) \stackrel{[X]}{\Longrightarrow} (s', \psi') \text{ in } \mathcal{P}$$

when there is a trail in $\mathcal{P}$ from (s, ψ) to (s', ψ') in which X is *active and unfolded*.

5.2.1 Trail Equivalence

Suppose $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a pre-model, and suppose (s, t) and (s', t') are some pairs of states in $\mathcal{P}$ such that $\Delta(s) = \Delta(s')$ and $\Delta(t) = \Delta(t')$. Roughly speaking, (s, t) is considered *trail-equivalent* to (s', t') if, for any formulae ψ, γ, whenever there is a trail from (s, ψ) to (t, γ), there is a trail from (s', ψ) to (t', γ), and vice versa. But for this notion to be useful, we need to take into account the variables which are active in such

trails. Precisely, the definitions of *trail equivalence* and its sister *trail inclusion* are as follows.

Definition 5.6. Suppose $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a pre-model. A pair (s,t) of states is *trail-included* by a pair (s',t'), written $(s,t) \sqsubseteq (s',t')$, iff

- $\Delta(s) = \Delta(s')$, $\Delta(t) = \Delta(t')$, and
- for each μ-variable Z and formulae $\psi \in \Delta(s), \gamma \in \Delta(t)$,
  - $(s,\psi) \stackrel{Z}{\Longrightarrow} (t,\gamma)$ implies that either $(s',\psi) \stackrel{Z}{\Longrightarrow} (t',\gamma)$ or $(s',\psi) \stackrel{[Z']}{\Longrightarrow} (t',\gamma)$, for some $Z' \prec Z$; and
  - $(s,\psi) \stackrel{[Z]}{\Longrightarrow} (t,\gamma)$ implies that $(s',\psi) \stackrel{[Z']}{\Longrightarrow} (t',\gamma)$, for some $Z' \preccurlyeq Z$.

(s,t) is *trail-equivalent* to (s',t'), written $(s,t) \cong (s',t')$, iff $(s,t) \sqsubseteq (s',t')$ and $(s',t') \sqsubseteq (s,t)$.

Based on these relations (on state pairs), we define some relations on states which are later used for model manipulation.

Definition 5.7. For any state s, define the relations $\sqsubseteq^s$ and $\cong^s$ on states:

- $t \sqsubseteq^s t'$ iff $(s,t) \sqsubseteq (s,t')$.
- $t \cong^s t'$ iff $(s,t) \cong (s,t')$.

It is quite clear that $\sqsubseteq$ and $\sqsubseteq^s$ are quasi-orderings (i.e. reflexive and transitive relations), and $\cong$ and $\cong^s$ are equivalence relations. More importantly, the numbers of equivalence classes of $\cong$ and $\cong^s$ are finite.

Proposition 5.8.

- *$\sqsubseteq$ and $\sqsubseteq^s$, for any state s, are quasi-orderings.*
- *$\cong$ and $\cong^s$, for any state s, are equivalence relations with $\leq 2^{2|\mu \mathrm{Var}(\phi)||\phi|^2}$ equivalence classes.*

Proof. Obviously, $\sqsubseteq$ and $\sqsubseteq^s$ are reflexive. To show transitivity, suppose $(s_1,t_1) \sqsubseteq (s_2,t_2)$ and $(s_2,t_2) \sqsubseteq (s_3,t_3)$. Obviously, we have $\Delta(s_1) = \Delta(s_3)$ and $\Delta(t_1) = \Delta(t_3)$. For any formulae ψ, γ and μ-variable Z,

$$
\begin{aligned}
& (s_1,\psi) \stackrel{Z}{\Longrightarrow} (t_1,\gamma), \\
\Rightarrow\ & \text{either } (s_2,\psi) \stackrel{Z}{\Longrightarrow} (t_2,\gamma) \text{ or } (s_2,\psi) \stackrel{[Z']}{\Longrightarrow} (t_2,\gamma), \text{ for some } Z' \prec Z, \\
\Rightarrow\ & \text{either } (s_3,\psi) \stackrel{Z}{\Longrightarrow} (t_3,\gamma) \text{ or } (s_3,\psi) \stackrel{[Z'']}{\Longrightarrow} (t_3,\gamma), \text{ for some } Z'' \prec Z.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
& (s_1,\psi) \stackrel{[Z]}{\Longrightarrow} (t_1,\gamma), \\
\Rightarrow\ & (s_2,\psi) \stackrel{[Z']}{\Longrightarrow} (t_2,\gamma), \text{ for some } Z' \preccurlyeq Z, \\
\Rightarrow\ & (s_3,\psi) \stackrel{[Z'']}{\Longrightarrow} (t_3,\gamma), \text{ for some } Z'' \preccurlyeq Z' \preccurlyeq Z.
\end{aligned}
$$

Hence, $\sqsubseteq$ is transitive, and clearly so is $\sqsubseteq^s$. This immediately implies that $\cong$ and $\cong^s$ are equivalence relations.

It is clear that the number of equivalence classes of $\cong$ or $\cong^s$ is no greater than the number of subsets of $\mathrm{Sub}(\phi) \times \mathrm{Sub}(\phi) \times (\mu\mathrm{Var}(\phi) \cup \{[Z] \mid Z \in \mu\mathrm{Var}(\phi)\})$, which is $\leq 2^{2|\mu\mathrm{Var}(\phi)||\phi|^2}$. □

Let us first consider some simple use of the above relations for model surgery. Suppose we have a tree pre-model $\mathcal{P}$ rooted at s, and a pair of non-root states t, t', where t' is below t, such that $t \sqsubseteq^s t'$. Consider the tree pre-model $\mathcal{P}' = \mathrm{Sub}_s(\mathrm{Jump}_{t,t'}(\mathcal{P}))$, i.e. $\mathcal{P}'$ is the result of replacing the subtree rooted at t in $\mathcal{P}$ by the one rooted at t'. Suppose there is a trail τ in $\mathcal{P}'$ from (s, ψ) to (t'', γ), for some descendant t'' of t', in which a μ-variable is active. We claim that there must be a similar trail in $\mathcal{P}$. The trail τ in $\mathcal{P}'$ must be of the form

$$\underbrace{(s,\psi) \to \ldots \to (s', [\![a]\!]\psi')}_{\tau_1} \to \underbrace{(t',\psi') \to \ldots \to (t'',\gamma)}_{\tau_2}$$

where $[\![a]\!]\psi'$ is some modal formula $\langle a \rangle \psi'$ or $[a]\psi'$. Hence, from the definition of $\mathrm{Jump}_{t,t'}(\mathcal{P})$,

$$\underbrace{(s,\psi) \to \ldots \to (s', [\![a]\!]\psi')}_{\tau_1} \to (t, \psi')$$

is a trail in $\mathcal{P}$. Since $t \sqsubseteq^s t'$, there must be a trail τ' in $\mathcal{P}$ from (s, ψ) to (t', ψ') in which some μ-variable is active. This implies that $\tau' \cdot \tau_2$ is a trail in $\mathcal{P}$ from (s, ψ) to (t'', γ) (in which some μ-variable is active).

If $\mathcal{P}$ is finite, then $\mathcal{P}'$ is "compatible" with $\mathcal{P}$ in the sense that, for each trail in $\mathcal{P}'$ from the root s to a leaf t'' in which a μ-variable is active, there is a similar trail from s to t'' in $\mathcal{P}$. Since $\mathcal{P}'$ is smaller than $\mathcal{P}$, by repeating this operation to all pairs of states t, t' where $t \sqsubseteq^s t'$, we obtain a tree pre-model whose height is no greater than the number of equivalence classes of $\cong^s$. This operation will later be used in the proof of the small model theorem for Π_2^μ-formulae.

The above argument can be generalised to arbitrary pre-models and any jump f, where $t \sqsubseteq^s f(t)$ for all states t, as follows.

Lemma 5.9. *Suppose $\mathcal{P} = \langle \mathcal{S}, \Delta, \to \rangle$ is a pre-model. For any state s, if f is a jump on $\mathcal{P}$ such that $t \sqsubseteq^s f(t)$ for each state t, then for each state s', formulae $\psi \in \Delta(s)$, $\psi' \in \Delta(s')$, and μ-variable Z,*

(a) $(s,\psi) \stackrel{Z}{\Longrightarrow} (s',\psi')$ in $\mathrm{Jump}_f(\mathcal{P})$ implies that either $(s,\psi) \stackrel{Z}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$ or $(s,\psi) \stackrel{[Z']}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$ for some $Z' \prec Z$; and

(b) $(s,\psi) \stackrel{[Z]}{\Longrightarrow} (s',\psi')$ in $\mathrm{Jump}_f(\mathcal{P})$ implies that $(s,\psi) \stackrel{[Z']}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$, for some $Z' \preccurlyeq Z$.

Proof. Suppose f is a jump on $\mathcal{P}$ such that $t \sqsubseteq^s f(t)$ for each state t.

(a) We prove by induction on $n \geq 1$ that if there is a trail τ of length n from (s, ψ) to (s', ψ') in which a μ-variable Z is active in $\text{Jump}_f(\mathcal{P})$, then either $(s, \psi) \stackrel{Z}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ or $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ for some $Z' \prec Z$.

For the basis step ($n = 1$), τ is the trail $(s, \psi) \rightarrow (s', \psi')$ in $\text{Jump}_f(\mathcal{P})$. From the definition of $\text{Jump}_f(\mathcal{P})$, if ψ is *not* a modal formula, then $(s, \psi) \rightarrow (s', \psi')$ is a trail in $\mathcal{P}$, and we are done. Otherwise, there must be a state t such that $s' = f(t)$ and $(s, \psi) \rightarrow (t, \psi')$ is a trail in $\mathcal{P}$. Since $t \sqsubseteq^s s'$, this implies that either $(s, \psi) \stackrel{Z}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ or $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ for some $Z' \prec Z$. Hence the hypothesis holds for $n = 1$.

Assume that the hypothesis holds for any trail of length n. Consider a trail τ in $\text{Jump}_f(\mathcal{P})$ of length $n + 1$ in which Z is active:

$$(s, \psi) = (s_0, \psi_0) \rightarrow ... \rightarrow (s_n, \psi_n) \rightarrow (s', \psi'),$$

From the hypothesis, there are two possible cases:

(1) $(s, \psi) \stackrel{Z}{\Longrightarrow} (s_n, \psi_n)$ in $\mathcal{P}$.
(2) $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (s_n, \psi_n)$ in $\mathcal{P}$ for some $Z' \prec Z$.

If ψ_n is *not* a modal formula, then

$$(s_n, \psi_n) \rightarrow (s', \psi')$$

is a trail in $\mathcal{P}$. For case (1), we have $(s, \psi) \stackrel{Z}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$. For case (2), we have $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$, for some $Z' \prec Z$.

If ψ_n is a modal formula, there must be a state t such that $s' = f(t)$ and

$$(s_n, \psi_n) \rightarrow (t, \psi')$$

is a trail in $\mathcal{P}$. For case (1), we have $(s, \psi) \stackrel{Z}{\Longrightarrow} (t, \psi')$ in $\mathcal{P}$. Since $t \sqsubseteq^s f(t) = s'$, this implies that either $(s, \psi) \stackrel{Z}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ or $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$ for some $Z' \prec Z$. For case (2), we have $(s, \psi) \stackrel{[Z']}{\Longrightarrow} (t, \psi')$ in $\mathcal{P}$, for some μ-variable $Z' \prec Z$. Since $t \sqsubseteq^s s'$, this implies that $(s, \psi) \stackrel{[Z'']}{\Longrightarrow} (s', \psi')$ in $\mathcal{P}$, for some μ-variable $Z'' \preccurlyeq Z'$.

Thus the hypothesis holds for any trail of length $n + 1$ and, therefore, (a) is true.

(b) Suppose τ is a trail in $\text{Jump}_f(\mathcal{P})$ from (s, ψ) to (s', ψ') in which Z is active and unfolded. Thus τ is of the form

$$\underbrace{(s, \psi) \rightarrow ... \rightarrow (t, Z)}_{\tau_1} \rightarrow \underbrace{(t, \psi) \rightarrow ... \rightarrow (s', \psi')}_{\tau_2}.$$

From the definition of $\text{Jump}_f(\mathcal{P})$, $(t, Z) \rightarrow (t, \psi)$ is also a trail in $\mathcal{P}$. Consider the trails τ_1 and τ_2. Assume that τ_1 and τ_2 are of length ≥ 1. By part (a), we have

- either $(s, \psi) \stackrel{Z}{\Longrightarrow} (t, Z)$ in $\mathcal{P}$ or $(s, \psi) \stackrel{[Z_1]}{\Longrightarrow} (t, Z)$ in $\mathcal{P}$ for some $Z_1 \prec Z$; and

- either $(t,\psi) \stackrel{Z}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$ or $(t,\psi) \stackrel{[Z_2]}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$ for some $Z_2 \prec Z$.

We can thus deduce that $(s,\psi) \stackrel{[Z']}{\Longrightarrow} (s',\psi')$ in $\mathcal{P}$, for some $Z' \preccurlyeq Z$. The case where τ_1 or τ_2 has length 0 can be shown similarly. □

Note that if $\mathcal{P}$ is well-founded, $\mathrm{Jump}_f(\mathcal{P})$, where $t \sqsubseteq^s f(t)$ for all states t, is *not* necessarily well-founded. All we can say is that there will be no μ-trail in $\mathrm{Jump}_f(\mathcal{P})$ which goes through s infinitely often.

Lemma 5.10. *Suppose $\mathcal{P}$ is a pre-model. For any state s, if f is a jump such that $t \sqsubseteq^s f(t)$ for every state t, then if* $\mathrm{Jump}_f(\mathcal{P})$ *contains a μ-trail which goes through s* infinitely often*, then so does $\mathcal{P}$.*

Proof. Suppose s is a state and f is a jump such that $t \sqsubseteq^s f(t)$ for each state t. Assume that there is a μ-trail τ in $\mathrm{Jump}_f(\mathcal{P})$ which goes through s infinitely often. Hence there must be an infinite sequences (s,ψ_1), (s,ψ_2), ... along the trail τ such that, for some μ-variable Z,

$$(s,\psi_1) \stackrel{[Z]}{\Longrightarrow} (s,\psi_2) \stackrel{[Z]}{\Longrightarrow} ...$$

in $\mathrm{Jump}_f(\mathcal{P})$. By the previous lemma, we may infer that

$$(s,\psi_1) \stackrel{[Z_1]}{\Longrightarrow} (s,\psi_2) \stackrel{[Z_2]}{\Longrightarrow} ...$$

in $\mathcal{P}$, where each Z_i is a μ-variable and $Z_i \preccurlyeq Z$. Let Z_n be the highest variable occurring infinitely often in the sequence $Z_1, Z_2, ...$ It follows that there is a trail in $\mathcal{P}$ which goes through s infinitely often and Z_n is active and unfolded infinitely often. □

Proposition 5.11. *Suppose $\mathcal{P}$ is a* well-founded *pre-model. For any state s, if f is a jump such that $t \sqsubseteq^s f(t)$ for every state t, then* $\mathrm{Jump}_f(\mathcal{P})$ *does not contain a μ-trail which goes through s infinitely often.*

Proof. A direct consequence of the previous lemma. □

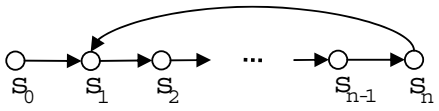


Figure 5.2: Pre-model $\mathcal{P}$ in Example 5.12. If $s_i \sqsubseteq^{s_1} s_j$ (where $1 < i < j \leq n$), then the region between s_i and s_j can be safely removed.

Example 5.12. Suppose $\mathcal{P}$ is a pre-model which consists of states $s_0, s_1, ..., s_n$ (for some $n > 1$), where all the transitions are as follows: $s_i R_a s_{i+1}$, for each $i < n$, and $s_n R_a s_1$ (see Figure 5.2). Suppose there is a pair of states s_i, s_j $(1 < i < j \leq n)$ such

that $s_i \sqsubseteq^{s_1} s_j$. Clearly, every infinite trail in $\text{Jump}_{s_i,s_j}(\mathcal{P})$ must go through s_1 infinitely often. Hence, by Proposition 5.11, if $\mathcal{P}$ is well-founded, then so is $\text{Jump}_{s_i,s_j}(\mathcal{P})$.

5.2.2 Safe Pairs of States

On the quest to prove the small model theorem, one of the problems we looked at is to identify the condition on a pair of states s, t along a well-founded *linear* pre-model (or, more generally, a branch of a tree pre-model) which guarantees that, when adding a backedge from t to s, the resulting pre-model is still well-founded. A more general question is the following: for any well-founded pre-model $\mathcal{P}$ and states s, t, what is the condition on s, t which guarantees that $\text{Jump}_{t,s}(\mathcal{P})$ is well-founded? And does a pair of states satisfying such condition always exist if the pre-model is *large* enough? This has led us to the notion of *safe pairs* of states.

Definition 5.13 (Safe Pairs)**.** Suppose $\mathcal{P}$ is a pre-model. For any states s, t with the same annotation, (s, t) is said to be *unsafe* (in $\mathcal{P}$) iff there exist a μ-variable Z and formulae $\psi_1, ..., \psi_n$ $(n \geq 1)$ such that

$$(s, \psi_n) \stackrel{[Z]}{\Longrightarrow} (t, \psi_1), \text{ and}$$
$$(s, \psi_i) \stackrel{Z}{\Longrightarrow} (t, \psi_{i+1}), \text{ for each } i,\ 1 \leq i < n;$$

(s, t) is said to be *safe* (in $\mathcal{P}$) otherwise.

Proposition 5.14. *Let (s, t) be a pair of states with the same annotation in a pre-model $\mathcal{P}$. If $\mathcal{P}$ is well-founded and (s, t) is safe in $\mathcal{P}$, then $\text{Jump}_{t,s}(\mathcal{P})$ is well-founded.*

Proof. Suppose (s, t) is safe in $\mathcal{P}$. Assume that $\text{Jump}_{t,s}(\mathcal{P})$ is *not* well-founded. Hence there is a trail τ in $\text{Jump}_{t,s}(\mathcal{P})$ in which a μ-variable Z is active and unfolded infinitely often. Since $\mathcal{P}$ is well-founded, τ (or any suffix of it) is not a trail in $\mathcal{P}$. It follows from the definition of $\text{Jump}_{t,s}(\mathcal{P})$ that τ must be of the form

$$... \to (s_1, \gamma_1) \to \underbrace{(s, \psi_1) \to ... \to (s_2, \gamma_2)}_{\tau_1} \to \underbrace{(s, \psi_2) \to ... \to (s_3, \gamma_3)}_{\tau_2} \to (s, \psi_3) \to ...$$

where, for all $i \geq 1$, τ_i and $(s_i, \gamma_i) \to (t, \psi_i)$ are trails in $\mathcal{P}$. This implies that, for each $i \geq 1$,

$$(s, \psi_i) \stackrel{Z}{\Longrightarrow} (t, \psi_{i+1}) \text{ in } \mathcal{P}.$$

Since Z is unfolded infinitely often in τ, for infinitely many j,

$$(s, \psi_j) \stackrel{[Z]}{\Longrightarrow} (t, \psi_{j+1}) \text{ in } \mathcal{P}.$$

Since there are finitely many formulae annotating $\mathcal{P}$, this clearly implies that (s, t) is unsafe in $\mathcal{P}$, contradicting the assumption. Hence $\text{Jump}_{t,s}(\mathcal{P})$ must be well-founded.

□

In other words, if (s,t) is safe in $\mathcal{P}$, we may jump from t to s without losing well-foundedness. We then ask the following: if (s_1,t_1) and (s_2,t_2) are safe in $\mathcal{P}$, is it safe to *simultaneously* jump from t_1 to s_1 and from t_2 to s_2 (i.e. whether $\mathrm{Jump}_f(\mathcal{P})$, where $f(t_1)=s_1$, $f(t_2)=s_2$, and $f(s)=s$ for other state s, is still well-founded)? Unfortunately, this is *not* the case. For example, imagine a well-founded pre-model with distinct states s,t,t' such that both (s,t) and (s,t') are safe, but there is a trail from (s,ψ_1) to (t,ψ_2) and a trail from (s,ψ_2) to (t',ψ_1) (for some formulae ψ_1,ψ_2) where a μ-variable Z is active in both trails and Z is unfolded in one of the trails. Clearly, adding backedges from both t and t' to s will introduce a μ-trail.

Next, we will show that every *infinite* well-founded pre-model must contain a safe pair. The proof of this involves an interesting combinatorial argument. First, we note that the notion of safe pairs is invariant under trail equivalence $\cong$.

Lemma 5.15. *If (s,t) is safe and $(s,t)\cong(s',t')$, then (s',t') is safe.*

Proof. This follows easily from the definition. □

Lemma 5.16. *Let $s,s_1,s_2,\ldots$ be any states with the same annotation in a* well-founded *pre-model.*

(a) (s,s) is safe.
(b) If $(s_1,s_2)\cong\ldots\cong(s_{n-1},s_n)\cong(s_n,s_1)$ ($n\geq 2$), then all pairs (s_i,s_{i+1}), $1\leq i<n$, and (s_n,s_1) are safe.
(c) If $(s_1,s_2)\cong(s_2,s_3)\cong\ldots$, then all pairs (s_i,s_{i+1}), $i\geq 1$, are safe.

Proof. We explain (c) only. Suppose (s_1,s_2) is an unsafe pair in a well-founded pre-model $\mathcal{P}$. Thus, for some μ-variable Z and formulae $\psi_1,\ldots,\psi_n$ ($n\geq 1$),

$$(s_1,\psi_n)\stackrel{[Z]}{\Longrightarrow}(s_2,\psi_1),\ \text{and}$$
$$(s_1,\psi_i)\stackrel{Z}{\Longrightarrow}(s_2,\psi_{i+1}),\ \text{for each } i,\ 1\leq i<n.$$

Since $(s_1,s_2)\cong(s_2,s_3)\cong\ldots$, we have

$$(s_1,\psi_1)\stackrel{Z_1}{\Longrightarrow}(s_2,\psi_2)\stackrel{Z_2}{\Longrightarrow}\ldots\stackrel{Z_{n-1}}{\Longrightarrow}(s_n,\psi_n)\stackrel{[Z_n]}{\Longrightarrow}(s_{n+1},\psi_1),$$

for some μ-variable $Z_i\preccurlyeq Z$, $i\geq 1$. This clearly implies that $(s_1,\psi_1)\stackrel{Z'}{\Longrightarrow}(s_{n+1},\psi_1)$ for some μ-variable $Z'\preccurlyeq Z$.

We may repeat this argument to obtain $(s_{kn+1},\psi_1)\stackrel{[Z'_k]}{\Longrightarrow}(s_{(k+1)n+1},\psi_1)$, for some μ-variable Z'_k and each $k\geq 0$. But this is impossible because $\mathcal{P}$ is well-founded. Hence (s_1,s_2) and, by the previous lemma, all pairs (s_i,s_{i+1}) are safe. □

The following property follows from Ramsey Theorem.

Lemma 5.17. *Every infinite sequence $s_1, s_2, ...$ of states must contain an infinite subsequence $s_{n_1}, s_{n_2}, ...$ (where $n_i < n_{i+1}$ for each i) such that $(s_{n_i}, s_{n_j}) \cong (s_{n_{i'}}, s_{n_{j'}})$, for all $i < j$ and $i' < j'$.*

Proof. Let $C = \{(i,j) \mid 1 \leq i < j\}$. Since there are finitely many $\cong$-equivalence classes, we can partition C into $C_1, ..., C_n$ where, for each $k \leq n$, if (i,j) and (i',j') are in C_k, then $(s_i, s_j) \cong (s_{i'}, s_{j'})$. By Ramsey Theorem (Theorem 0.8), there must be some $k \leq n$ and an infinite sequence $n_1 < n_2 < ...$ such that (n_i, n_j) is in C_k for all $i < j$. This means that $(s_{n_i}, s_{n_j}) \cong (s_{n_{i'}}, s_{n_{j'}})$, for all $i < j$ and $i' < j'$. □

Proposition 5.18. *Every infinite sequence $s_1, s_2, ...$ of states in a* well-founded *pre-model $\mathcal{P}$ must contain an infinite subsequence $s_{n_1}, s_{n_2}, ...$ (where $n_i < n_{i+1}$ for each i) such that (s_{n_i}, s_{n_j}) is safe in $\mathcal{P}$ for all $i < j$.*

Proof. By Lemma 5.17, there is an infinite subsequence $s_{n_1}, s_{n_2}, ...$ such that $(s_{n_i}, s_{n_j}) \cong (s_{n_{i'}}, s_{n_{j'}})$, for all $i < j$ and $i' < j'$. By Lemma 5.16, all the pairs (s_{n_i}, s_{n_j}), $i < j$, are safe in $\mathcal{P}$. □

5.2.3 Small Model Theorem: Linear Case

By a *linear model*, we mean an *infinite* tree model of degree 1. Using the result in the previous section, it is not difficult to show that every formula with a linear model has a small model. The idea is that, in any a well-founded linear pre-model $\mathcal{P}$, there must be a safe pair (s,t) of states (where t is below s). By Proposition 5.14, we can safely add a backedge from t to s to obtain a finite well-founded pre-model. We then perform some pruning on this finite pre-model to obtain a small pre-model, as done in Example 5.12.

Theorem 5.19. *If ϕ has a linear model, then it has a finite model with $\leq 2^{2|\mu \mathrm{Var}(\phi)||\phi|^2}$ states.*

Proof. Suppose $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}} \rangle$ is a linear model such that ϕ is true at the root state s_0. By Theorem 3.27, there exists a dependency relation $\rightarrow$ such that the pre-model $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$, where Δ is the canonical annotation of the subformulae of ϕ on $\mathcal{M}$, is well-founded.

Suppose $s_0, s_1, ...$ is the sequence of (consecutive) states in the model. By Proposition 5.18, there is a pair of states s_m, s_n $(0 \leq m < n)$ such that (s_m, s_n) is safe in $\mathcal{P}$. By Proposition 5.14, the pre-model $\mathcal{P}_0 = \mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_n, s_m}(\mathcal{P}))$ is well-founded. $\mathcal{P}_0$ is an eventually-cyclic structure, as shown in Figure 5.3. Call the states $s_0, ..., s_{m-1}$ the *prefix* part, and the states $s_m, ..., s_{n-1}$ the loop part.

We reduce the size of the loop part by removing the section between each pair of $\cong^{s_m}$-equivalent states. Precisely, let s_i, s_j, where $m < i < j \leq n-1$, be states in the loop part such that $s_i \cong^{s_m} s_j$. Then, by Lemma 5.9, $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_i, s_j}(\mathcal{P}_0))$ is still well-founded. Successively repeat this operation until no such pair of states exist. Call

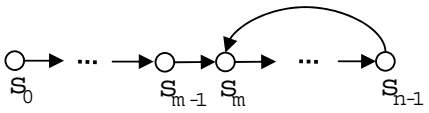


Figure 5.3: Pre-model $\mathrm{Sub}_{s_0}(\mathrm{Jump}_{s_n,s_m}(\mathcal{P}))$

the resulting (well-founded) pre-model $\mathcal{P}'$. The loop part of $\mathcal{P}'$ is a subsequence of $s_m, ..., s_{n-1}$, whereas the prefix part is $s_0, ..., s_{m-1}$ as before.

The next step is to reduce the size of the prefix part by eliminating the states on the prefix part which has the same annotation as some state in the loop part (skip this step if there is no such state). Suppose s_i is the first such state, and s_j be some state in the loop part with same annotation as s_i. Let $\mathcal{P}''$ be the pre-model $\mathrm{Jump}_{s_i,s_j}(\mathcal{P}')$ with all the states $s_i, ..., s_{m-1}$ removed. Hence, $\mathcal{P}''$ is an eventually-cyclic structure as shown in Figure 5.4. Notice that, if $s_i = s_0$ then the prefix part is now empty.

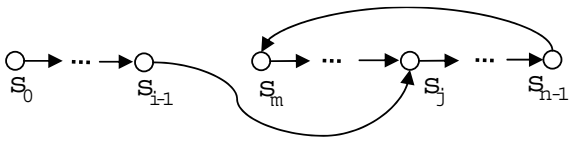


Figure 5.4: Pre-model $\mathcal{P}''$

It is clear that every infinite trail in $\mathcal{P}''$ must contain a suffix which is a trail in $\mathcal{P}'$. Hence $\mathcal{P}''$ must be well-founded. Since ϕ annotates some state in $\mathcal{P}''$, by the Fundamental Semantic Theorem, any model based on $\mathcal{P}''$ satisfies ϕ.

Since the loop part of $\mathcal{P}''$ does not contain distinct $\cong^{s_m}$-equivalent states and the prefix part of $\mathcal{P}''$ does not contain a state with the same annotation as some state in the loop part, it follows that the number of states of $\mathcal{P}''$ is no greater than the number of $\cong^{s_m}$-equivalent classes, which (by Proposition 5.8) is $\leq 2^{2|\mu\mathrm{Var}(\phi)||\phi|^2}$. □

5.3 Π_2^μ-Formulae

The technique studied in the previous section can be used to prove a small model theorem for the formulae in Π_2^μ. Recall from Proposition 2.30 that a formula ϕ (in positive normal form) is in Π_2^μ iff, for each pair of fixpoint subformulae $\mu X.\psi_X$ and $\nu Y.\psi_Y$ of ϕ, X does not occur free in ψ_Y. This means that a formula ϕ in Π_2^μ can have a ν/μ alternation, but *not* a μ/ν alternation, of fixpoint formulae. For this reason, formulae in Π_2^μ are sometimes called $\nu\mu$ *formulae*.

Example 5.20. The following formulae are in Π_2^μ:

$$\nu X.\langle a\rangle(\mu Y.X \vee [a]Y) \vee \langle a\rangle(\nu Z.X \vee [a]Z),$$
$$\mu Z.(\nu Y.Q \wedge [a]Y) \vee \langle a\rangle Z,$$
$$\nu Y.\mu X.[a]X \vee \nu Z.\langle a\rangle(Y \wedge Z),$$

while the following are not:

$$\mu X.\langle a\rangle(\nu Y.X \vee [a]Y) \vee \langle a\rangle(\mu Z.X \vee [a]Z),$$
$$\mu Z.\nu Y.\langle a\rangle Z.$$

Proposition 5.21. *For any formula ϕ in Π_2^μ, no μ-variable is active in a greatest-fixpoint subformula of ϕ.*

Proof. Suppose ϕ is a Π_2^μ-formula. By definition, if a variable Z is active in a formula $\nu Y.\psi$ in ϕ, there must be a sequence $X_0(=Z), X_1, ..., X_n$ such that

- each X_i occurs free in the formula $\sigma_{i+1}X_{i+1}.\psi_{i+1}$ identified by X_{i+1}, and
- X_n occurs free in $\nu Y.\psi$.

Since no μ-variable may occur free in a greatest fixpoint formula in ϕ, all the variables $X_0, ..., X_n$ must be ν-variables. □

5.3.1 Small Model Theorem for Π_2^μ

A pre-model annotated with subformulae of a Π_2^μ-formula has the following useful property. Suppose $\mathcal{P}$ is such a pre-model. First, observe that an infinite trail in $\mathcal{P}$ must contain infinitely many unfoldings of some variable, say X. Now if a μ-variable Z is active throughout such trail, obviously Z must be active in X. By Proposition 5.21, X must be a μ-variable. Thus we obtain the following lemma.

Lemma 5.22. *Suppose $\mathcal{P}$ is a pre-model annotated by subformulae of a Π_2^μ-formula. Every infinite trail in $\mathcal{P}$ in which some μ-variable is active must contain infinitely many unfoldings of a μ-variable.*

Proof. Consider an infinite trail τ along which a μ-variable Z is active. Assume that no μ-variable is unfolded infinitely often in this trail. Since there are only finitely many variables, there must be a suffix τ' of this trail along which no μ-variable is unfolded. Since the number of subformulae is finite, τ' must contain a subtrail of the form $(s, \psi) \to ... \to (s', \psi)$. Clearly some variable must be unfolded in this subtrail, and from our assumption it must be a ν-variable, say X. But since ϕ is in Π_2^μ, no μ-variable can be active in X or its unfolding. This contradicts the assumption that some μ-variable is active throughout. Hence the lemma is true. □

Active trail. For brevity, let us call a trail in which a μ-variable is active an *active trail*. Suppose $\mathcal{P}$ is a well-founded *tree* pre-model of finite degree for a Π^{μ}_2-formula ϕ. By the previous lemma, every active trail in $\mathcal{P}$ from a state s must be finite. Hence, by König's Lemma, there must be some level n in $\mathcal{P}$ such that there is no active trail from s to any state at level greater than n. In other words, for any state s in $\mathcal{P}$ there exists a finite partial subtree $\mathcal{T}_s$ rooted at s such that there is no active trail from s to any leaf t of $\mathcal{T}_s$, except when t is also a leaf of $\mathcal{P}$.

Terminal state. A state s in a tree pre-model $\mathcal{P}$ is said to be a *terminal state* iff s does not contain a $\langle\cdot\rangle$-formula in its annotation; s is said to be a *non-terminal state* otherwise. Obviously, every leaf of $\mathcal{P}$ is a terminal state. But an internal state of $\mathcal{P}$ could also be a terminal state. In such case, we may safely remove all the (proper) descendants of such terminal state from $\mathcal{P}$.

Proposition 5.23. *Let ϕ be a Π^{μ}_2-formula and $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ be a well-founded tree pre-model of* finite *degree annotated by subformulae of ϕ. For each state s of $\mathcal{P}$, there exists a* finite *partial subtree $\mathcal{T}_s$ of $\mathcal{S}$ rooted at s such that*

> *($\star$) for each leaf t of $\mathcal{T}_s$, there is no active trail from s to t, unless t is a terminal state.*

Proof. Define T to be the set of all descendants t of s in $\mathcal{P}$ such that there exists an active trail from s to t. Clearly, if t is in T, then so is every state between s and t.

We claim that T must be finite. Assume otherwise. Since $\mathcal{P}$ is assumed to be of finite degree, there must be an infinite path of states in T. Since there is an active trail from s to each state t in T, by König's lemma (and the fact that each state is annotated by finitely many formulae), there must be an infinite trail from s along this path in which a μ-variable is active. By the previous lemma, some μ-variable must be unfolded infinitely often along such trail. This contradicts the well-foundedness of $\mathcal{P}$. Hence T is finite.

Let T' contain all states in T and all the children of t, for each state $t \in T$ which contains a $\langle\cdot\rangle$-formula in its annotation. Define $\mathcal{T}_s$ to be the subsystem of $\mathcal{S}$ whose states are precisely T'. $\mathcal{T}_s$ is clearly a finite partial subtree of $\mathcal{S}$ rooted at s satisfying ($\star$) as required. □

Remark 5.24. By this proposition, given a well-founded tree pre-model $\mathcal{P}$ where ϕ annotates the root, it is easy to construct a finite model for ϕ. Suppose $\Gamma_1, ..., \Gamma_n \subseteq \mathrm{Sub}(\phi)$ are all the annotations of states in $\mathcal{P}$ (i.e. each Γ_i annotates some state s_i). By the proposition, for each $i \leq n$, there exists a finite partial subtree $\mathcal{T}_i$ of $\mathcal{P}$ satisfying ($\star$) whose root is annotated by Γ_i. Let $\mathcal{P}_i = \langle \mathcal{T}_i, \Delta_i, \rightarrow_i \rangle$ be the restriction of $\mathcal{P}$ to the states in $\mathcal{T}_i$. Since $\mathcal{P}$ is well-founded, there is no μ-trail in any $\mathcal{P}_i$. Now consider the disjoint union of $\mathcal{P}_1, ..., \mathcal{P}_n$. For each non-terminal leaf t of $\mathcal{P}_i$, if t is annotated by Γ_j, we identify t with the root of $\mathcal{P}_j$. We thus obtain a finite pre-model $\mathcal{P}'$. Since

there is no μ-trail within each $\mathcal{P}_i$ and no active trail from the root of $\mathcal{P}_i$ to any of its non-terminal leaves, there can be no μ-trail in the new pre-model $\mathcal{P}'$. Thus, by the Fundamental Semantic Theorem, any model based on $\mathcal{P}'$ satisfies ϕ.

The model obtained above, though finite, is unbound in size. This is because there is no bound on the size of the trees $\mathcal{T}_1, ..., \mathcal{T}_n$ used to construct the model. But this is easily fixed by performing some surgery on these trees first. The key is to ensure that $(\star)$ is preserved after the operation (i.e. there is no active trail from the root of $\mathcal{T}_i$ to any non-terminal leaf). One way is to prune these trees by eliminating equivalent pairs of states (e.g. applying Proposition 5.9 in the previous section). To obtain an optimal bound, we use a "coarser" version of trail equivalence.

Definition 5.25. Suppose $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ is a pre-model. For each state s, define the relation $\sqsubseteq^s_{\nu\mu}$ on S: $t \sqsubseteq^s_{\nu\mu} t'$ iff

- $\Delta(t) = \Delta(t')$, and
- if there is an active trail from (s, ψ) to (t, γ), then there must be an active trail from (s, ψ'), for some ψ', to (t', γ).

The relation $\cong^s_{\nu\mu}$ is defined thus: $t \cong^s_{\nu\mu} t'$ iff $t \sqsubseteq^s_{\nu\mu} t'$ and $t' \sqsubseteq^s_{\nu\mu} t$.

Proposition 5.26. *For each state s,*

- $\sqsubseteq^s_{\nu\mu}$ *is a quasi-ordering on states.*
- $\cong^s_{\nu\mu}$ *is an equivalence relation on states with* $\leq 3^{|\phi|}$ *equivalence classes.*

Proof. It is straightforward to check that $\sqsubseteq^s$ is a quasi-ordering and $\cong^s_{\nu\mu}$ is an equivalence relation on states. For any state s, there can be no more than $3^{|\phi|}$ pairwise $\cong^s_{\nu\mu}$-inequivalent states. To see this, consider any state t and subformulae ψ of ϕ. t and ψ fall into one of the following 3 categories:

(1) $\psi \notin \Delta(t)$.
(2) $\psi \in \Delta(t)$ and there is an active trail from s to (t, ψ).
(3) $\psi \in \Delta(t)$ but there is no active trail from s to (t, ψ).

For any pair t, t' of states, if for each subformula ψ of ϕ, t and ψ fall into the same category as t' and ψ, then clearly $t \cong^s_{\nu\mu} t'$. Hence there can be no more than $3^{|\phi|}$ states which are pairwise $\cong^s_{\nu\mu}$-inequivalent. □

We will later apply the following lemma to transform a finite tree pre-model into a small DAG-shaped pre-model.

Lemma 5.27. *Let $\mathcal{P}$ be a well-founded acyclic pre-model with root s_0. Suppose t, t' are distinct states such that there is no path from t' to t. If $t \sqsubseteq^{s_0}_{\nu\mu} t'$ then*

- $\mathrm{Jump}_{t,t'}(\mathcal{P})$ *is well-founded and acyclic, and*
- *for any state s and formula ψ, if* $\mathrm{Jump}_{t,t'}(\mathcal{P})$ *contains an active trail from s_0 to (s, ψ), then so does $\mathcal{P}$.*

Proof. Since there is no path from t' to t, it is clear that $\mathrm{Jump}_{t,t'}(\mathcal{P})$ is acyclic. Moreover, an infinite trail in $\mathrm{Jump}_{t,t'}(\mathcal{P})$ must contain a suffix which is trail in $\mathcal{P}$. Since $\mathcal{P}$ is well-founded, so is $\mathrm{Jump}_{t,t'}(\mathcal{P})$.

Suppose τ is an active trail in $\mathrm{Jump}_{t,t'}(\mathcal{P})$ from s_0 to some (s,ψ). Clearly, if τ does not go through t' then τ is also a trail in $\mathcal{P}$. Otherwise, τ must be of the form

$$\underbrace{(s_0,\psi_0) \to ... \to (s_n,\psi_n)}_{\tau_1} \to \underbrace{(t',\gamma) \to ... \to (s,\psi)}_{\tau_2}$$

where τ_1 and τ_2 are trails in $\mathcal{P}$. From the definition of $\mathrm{Jump}_{t,t'}(\mathcal{P})$, either $(s_n,\psi_n) \to (t',\gamma)$ in $\mathcal{P}$ or $(s_n,\psi_n) \to (t,\gamma)$ in $\mathcal{P}$. In the former case, τ is a trail in $\mathcal{P}$. In the latter case, $(s_0,\psi_0) \to ... \to (s_n,\psi_n) \to (t,\gamma)$ is a trail in $\mathcal{P}$. Since $t \sqsubseteq^{s_0}_{\nu\mu} t'$, this implies that there is an active trail from s_0 to (t',γ) in $\mathcal{P}$. Concatenating this latter trail with τ_2, we obtain an active trail from s_0 to (s,ψ) in $\mathcal{P}$. □

Theorem 5.28. *Any satisfiable formula ϕ in Π^μ_2 has a finite model with $\leq 6^{|\phi|}$ states.*

Proof. By Theorem 3.28, ϕ has a tree model $\mathcal{M} = \langle \mathcal{S}, \mathcal{V}_{\mathrm{Prop}} \rangle$ with finite degree. By Theorem 3.27, there exists a dependency relation $\to$ such that $\mathcal{P} = \langle \mathcal{S}, \Delta, \to \rangle$, where Δ is the canonical annotation of $\mathrm{Sub}(\phi)$ on $\mathcal{M}$, is a well-founded pre-model.

For each set $\Gamma \subseteq \mathrm{Sub}(\phi)$ annotating some state in $\mathcal{P}$, we select a state s_Γ which has Γ as its annotation. By Proposition 5.23, there exists a finite partial subtree $\mathcal{T}_\Gamma$ rooted at s_Γ such that

($\star$) for each leaf t of $\mathcal{T}_\Gamma$, there is no active trail from s_Γ to t, unless t is a terminal state.

Let $\mathcal{P}_0 = \langle \mathcal{T}_\Gamma, \Delta', \to' \rangle$ be the restriction of $\mathcal{P}$ to the states in $\mathcal{T}_\Gamma$. Since $\mathcal{P}$ is well-founded, $\mathcal{P}_0$ does not contain an infinite active trail.

We can reduce the size of $\mathcal{P}_0$ while preserving ($\star$) by iteratively applying Jump operation at each equivalent pair of states. Precisely, assume that after i iterations we obtain $\mathcal{P}_0, ...\mathcal{P}_i$ each of which is *acyclic* and rooted at s_Γ. Select two distinct non-root states t, t' in $\mathcal{P}_i$ such that $t \cong^{s_\Gamma}_{\nu\mu} t'$ in $\mathcal{P}_i$. Since $\mathcal{P}_i$ is acyclic, we may assume without loss of generality that there is no path from t' to t. Let

$$\mathcal{P}_{i+1} \quad = \quad \mathrm{Sub}_{s_\Gamma}(\mathrm{Jump}_{t,t'}(\mathcal{P}_i)).$$

By Lemma 5.27, $\mathcal{P}_{i+1}$ is acyclic and well-founded. Further, ($\star$) still holds because if there is a trail from s_Γ to a state s in which a μ-variable is active in $\mathcal{P}_{i+1}$ (and hence in $\mathrm{Jump}_{t,t'}(\mathcal{P}_i)$), there must be such a trail in $\mathcal{P}_i$. Since $\mathcal{P}_{i+1}$ is strictly smaller that $\mathcal{P}_i$, after some finite iterations, we obtain $\mathcal{P}_\Gamma$ in which no two distinct (non-root) states are $\cong^{s_\Gamma}_{\nu\mu}$-equivalent. By Lemma 5.26, $\mathcal{P}_\Gamma$ contains $\leq 3^{|\phi|}$ states.

It is now easy to construct a small model for ϕ. Let $\Gamma_1, ..., \Gamma_n$ be all the annotations of the states of $\mathcal{P}$, and $\mathcal{P}_{\Gamma_1}, ..., \mathcal{P}_{\Gamma_n}$ be the finite pre-models constructed above. Assume

that we have made the set of states of these $\mathcal{P}_{\Gamma_i}$ disjoint. Define a jump f as follows: for each non-terminal leaf t in $\mathcal{P}_{\Gamma_1}, ..., \mathcal{P}_{\Gamma_n}$, if t is annotated by Γ, $f(t)$ is the root of $\mathcal{P}_\Gamma$; for any other state s, $f(s) = s$. Let

$$\mathcal{P}' \quad = \quad \text{Jump}_f(\biguplus_{1 \leq i \leq n} \mathcal{P}_{\Gamma_i}).$$

Figure 5.5 illustrates how the pre-model $\mathcal{P}'$ is formed from the small pre-models $\mathcal{P}_{\Gamma_i}$.

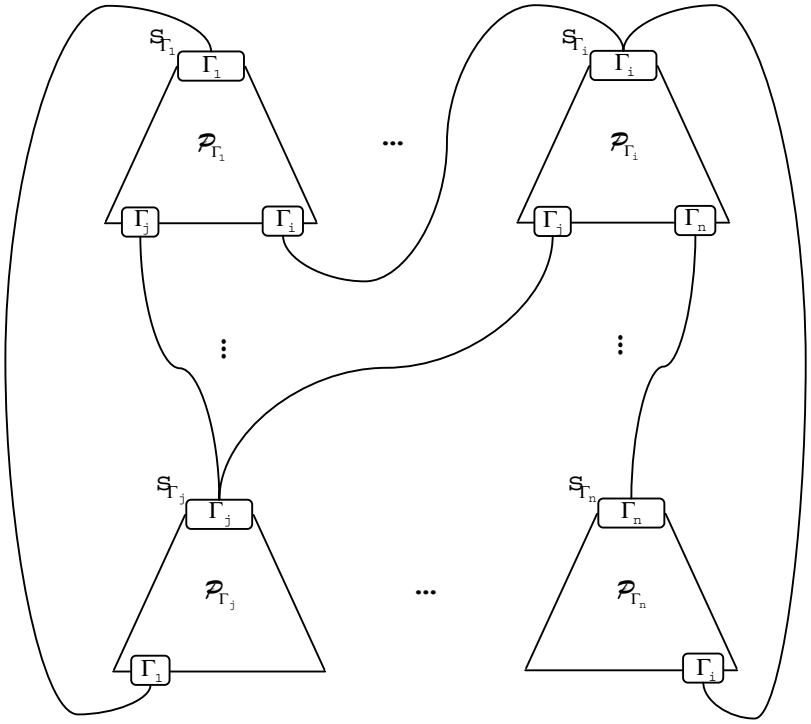


Figure 5.5: The small acyclic pre-models $\mathcal{P}_{\Gamma_i}$ (for all annotating sets Γ_i) are joined together to form a model for ϕ. For each non-terminal leaf s of $\mathcal{P}_{\Gamma_i}$, if s is annotated by Γ_j, we identify s with the root of $\mathcal{P}_{\Gamma_j}$.

We claim that $\mathcal{P}'$ is well-founded. Suppose there is an infinite μ-trail τ in $\mathcal{P}'$. Since each $\mathcal{P}_{\Gamma_i}$ is well-founded, τ and all its suffixes are not trails in $\mathcal{P}_{\Gamma_i}$. Hence, τ must be a concatenation of finite trails in $\mathcal{P}_{\Gamma_1}, ..., \mathcal{P}_{\Gamma_n}$, i.e. τ is of the form

$$\underbrace{(s, \psi) \to ... \to (s_1, \gamma_1)}_{\tau_1} \to \underbrace{(s_{\Gamma_2}, \psi_2) \to ... \to (s_2, \gamma_2)}_{\tau_2} \to (s_{\Gamma_3}, \psi_3) \to ...$$

where, for each $i \geq 1$, τ_i is a trail in some $\mathcal{P}_{\Gamma'_i}$ and, for some non-terminal leaf t of $\mathcal{P}_{\Gamma'_i}$ annotated by Γ'_i, $(s_i, \gamma_i) \to (t, \psi_{i+1})$ in $\mathcal{P}_{\Gamma'_i}$. But by $(\star)$, no μ-variable can be active along τ, contradicting the assumption that τ is a μ-trail. Therefore, $\mathcal{P}'$ is well-founded. Hence, by Theorem 3.22, ϕ is satisfied in any model based on $\mathcal{P}'$.

Since there are $\leq 2^{|\phi|}$ possible annotations and each pre-model $\mathcal{P}_{\Gamma_i}$ contains $\leq 3^{|\phi|}$ states, the number of states in $\mathcal{P}'$ is no greater than $6^{|\phi|}$ states. □

5.3.2 Tableau System NUMU

The technique used in the proof of the small model theorem for Π_2^μ-formulae has led us to design a sound and complete tableau system for the fragment.

Let ϕ be a closed and guarded formula in Π_2^μ. In tableau system NUMU, a *goal* in a tableau for ϕ is a set Γ of *augmented formulae* of the form ψ^ρ, where $\rho \in \{0,1\}$. The initial goal is ϕ^0.

Tableau rules. The tableau rules of NUMU are as follows.

$$\mathsf{R}\wedge: \quad \frac{(\psi_1 \wedge \psi_2)^\rho, \Gamma}{\psi_1^\rho, \psi_2^\rho, \Gamma}$$

$$\mathsf{R}\vee: \quad \frac{(\psi_1 \vee \psi_2)^\rho, \Gamma}{\psi_i^\rho, \Gamma}, \quad i \in \{1,2\}$$

$$\mathsf{R}\sigma: \quad \frac{(\sigma X.\psi)^\rho, \Gamma}{X^\rho, \Gamma}, \quad \sigma \in \{\mu, \nu\}$$

$$\mathsf{Unfold}_\mu: \quad \frac{Z^\rho, \Gamma}{\psi^\rho, \Gamma}, \qquad Z \text{ identifies } \mu Z.\psi.$$

$$\mathsf{Unfold}_\nu: \quad \frac{X^\rho, \Gamma}{\psi^{\rho'}, \Gamma}, \qquad \begin{array}{l} X \text{ identifies } \nu X.\psi, \\ \rho' = 1 \text{ if there is } \textit{no } \mu\text{-variable active in } \psi, \\ \rho' = \rho \text{ otherwise.} \end{array}$$

$$\mathsf{R}\langle\rangle: \frac{(\langle a_1\rangle\psi_1)^{\rho_1}, ..., (\langle a_n\rangle\psi_n)^{\rho_n}, \Gamma}{\psi_1^{\rho_1}, \Gamma_{a_1} \mid ... \mid \psi_n^{\rho_n}, \Gamma_{a_n}}, n \geq 1,$$

where

- Γ contains only literals and $[\cdot]$-formulae,
- for each action a, $\Gamma_a = \{\psi^\rho \mid ([a]\psi)^\rho \in \Gamma\}$.

$$\mathsf{Shift}: \frac{\psi_1^1, ..., \psi_n^1, \Gamma}{\psi_1^0, ..., \psi_n^0, \Gamma}, \quad n \geq 1 \text{ and } \Gamma \text{ is a set of literals.}$$

$$\mathsf{Thin}: \frac{\psi^0, \psi^1, \Gamma}{\psi^0, \Gamma}$$

Remark 5.29. In rule Unfold_ν, ρ' is set to 1 if there is no μ-variable active in ψ. If the initial formula ϕ is a Π_2^μ (as we assumed) then the latter condition is always true (because no μ-variable is active in a greatest-fixpoint subformula of ϕ). This means

that, for Π_2^μ-formulae, rule Unfold_ν is equivalent to

$$\mathsf{Unfold}'_\nu : \quad \frac{X^\rho, \Gamma}{\psi^1, \Gamma}, \quad X \text{ identifies } \nu X.\psi.$$

The reason we stipulate the condition in rule Unfold_ν is to ensure that the tableau system is *sound* for all formulae in the modal μ-calculus. Without such condition (i.e. using Unfold'_ν instead), there will be an unsatisfiable (non-Π_2^μ) formula which has a successful tableau.

Note that rule Thin is *not* strictly necessary as there are only finitely many possible goals in a NUMU-tableau.

Termination. A *terminal* is a leaf u, labelled by a goal Γ, such that *one* of the following holds:

T1. Γ contains a complementary pair of literals.
T2. Γ is a consistent set of literals and $[\cdot]$-formulae.
T3. u has a proper ancestor v with the same goal Γ, called the *companion* of u.

Success. A *successful terminal* is a terminal u such that *one* of the following holds:

S1. The goal Γ at u satisfies T2.
S2. u has a companion v and rule Shift is applied between v and u.

A terminal is said to be *unsuccessful* otherwise.

A successful tableau T is a *finite* tableau in which all leaves are successful terminals.

Example 5.30. The Π_2^μ-formula

$$\nu X.(\mu Y.(P \wedge [a]X) \vee \langle a \rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a \rangle Z)$$

is satisfiable. Figure 5.6 shows a successful tableau for the formula. Node 28 is a successful terminal with node 6 as its companion. Rule Shift is applied at node 25.

Example 5.31. The formula

$$\nu X.\mu Z.((P \wedge [a]X) \vee \langle a \rangle Z) \wedge ((\neg P \wedge [a]X) \vee \langle a \rangle Z)$$

is unsatisfiable. The formula has no successful tableau. Figure 5.7 depicts a tree of unsuccessful tableaux for this formula (the other choice at node 8 obviously leads to an unsuccessful terminal).

1: $\dfrac{\nu X.(\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\nu$

2: $\dfrac{X^0}{} \; \mathsf{Unfold}_\nu$

3: $\dfrac{((\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z))^1}{} \; \mathsf{Shift}$

4: $\dfrac{((\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z))^0}{} \; \mathsf{R}\wedge$

5: $\dfrac{(\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y)^0,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\mu$

6: $\dfrac{Y^0,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\mu$

7: $\dfrac{Y^0,\ Z^0}{} \; \mathsf{Unfold}_\mu$

8: $\dfrac{((P \wedge [a]X) \vee \langle a\rangle Y)^0,\ Z^0}{} \; \mathsf{R}\vee$

9: $\dfrac{(P \wedge [a]X)^0,\ Z^0}{} \; \mathsf{R}\wedge$

10: $\dfrac{P^0,\ [a]X^0,\ Z^0}{} \; \mathsf{Unfold}_\mu$

11: $\dfrac{P^0,\ [a]X^0,\ ((\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\vee$

12: $\dfrac{P^0,\ [a]X^0,\ \langle a\rangle Z^0}{} \; \mathsf{R}\langle\rangle$

13: $\dfrac{X^0,\ Z^0}{} \; \mathsf{Unfold}_\nu$

14: $\dfrac{((\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z))^1,\ Z^0}{} \; \mathsf{R}\wedge$

15: $\dfrac{(\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y)^1,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^1,\ Z^0}{} \; \mathsf{R}\mu$

16: $\dfrac{Y^1,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^1,\ Z^0}{} \; \mathsf{R}\mu$

17: $\dfrac{Y^1,\ Z^1,\ Z^0}{} \; \mathsf{Thin}$

18: $\dfrac{Y^1,\ Z^0}{} \; \mathsf{Unfold}_\mu$

19: $\dfrac{(P \wedge [a]X) \vee \langle a\rangle Y)^1,\ Z^0}{} \; \mathsf{R}\vee$

20: $\dfrac{\langle a\rangle Y^1,\ Z^0}{} \; \mathsf{Unfold}_\mu$

21: $\dfrac{\langle a\rangle Y^1,\ ((\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\vee$

22: $\dfrac{\langle a\rangle Y^1,\ (\neg P \wedge [a]X)^0}{} \; \mathsf{R}\wedge$

23: $\dfrac{\langle a\rangle Y^1,\ \neg P^0,\ [a]X^0}{} \; \mathsf{R}\langle\rangle$

24: $\dfrac{Y^1,\ X^0}{} \; \mathsf{Unfold}_\nu$

25: $\dfrac{Y^1,\ ((\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z))^1}{} \; \mathsf{Shift}$

26: $\dfrac{Y^0,\ ((\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y) \wedge (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z))^0}{} \; \mathsf{R}\wedge$

27: $\dfrac{Y^0,\ (\mu Y.(P \wedge [a]X) \vee \langle a\rangle Y)^0,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{} \; \mathsf{R}\mu$

28: $\dfrac{Y^0,\ (\mu Z.(\neg P \wedge [a]X) \vee \langle a\rangle Z)^0}{}$

SUCCESSFUL

Figure 5.6: A successful NUMU-tableau for Example 5.30.

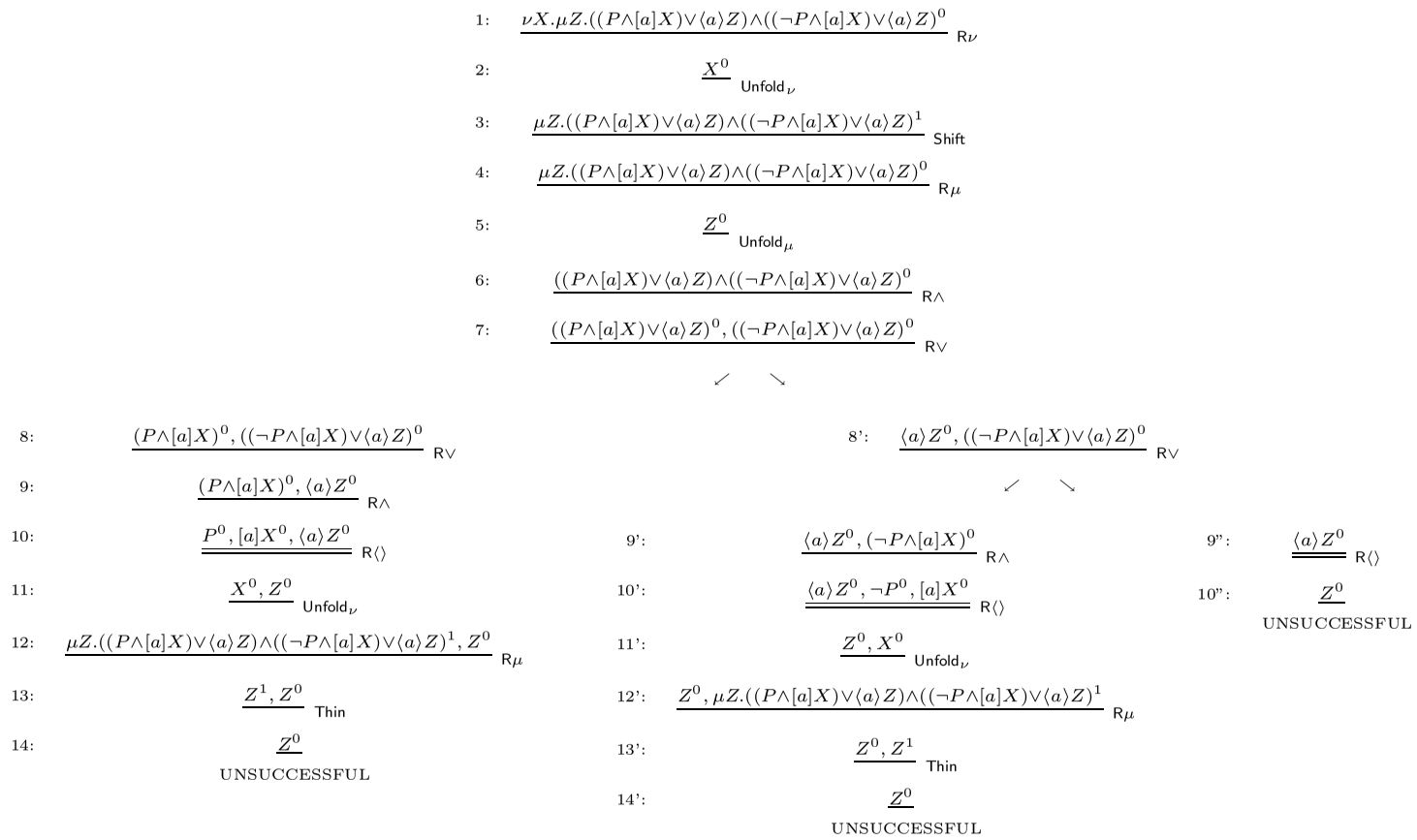


Figure 5.7: A tree of unsuccessful NUMU-tableaux for Example 5.31.

Example 5.32. The non-Π_2^μ formula

$$\mu Z.\nu Y.\langle a\rangle Z \vee ((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2)),$$

is satisfiable. The formula has no successful NUMU-tableau. As can be seen from the unsuccessful tableau in Figure 5.8, the unfolding of the ν-variable Y^0 at node 5 and 16 is *not* augmented with 1 because Z is active in it.

$$
\begin{array}{ll}
1: & \dfrac{\mu Z.\nu Y.\langle a\rangle Z \vee ((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0}{} \ \mathsf{R}\mu \\
2: & \dfrac{Z^0}{} \ \mathsf{Unfold}_\mu \\
3: & \dfrac{\nu Y.\langle a\rangle Z \vee ((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0}{} \ \mathsf{R}\nu \\
4: & \dfrac{Y^0}{} \ \mathsf{Unfold}_\nu \\
5: & \dfrac{\langle a\rangle Z \vee ((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0}{} \ \mathsf{R}\vee \\
6: & \dfrac{((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0}{} \ \mathsf{R}\wedge \\
7: & \dfrac{(\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1)^0,\ (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0}{} \ \mathsf{R}\mu \\
8: & \dfrac{X_1^0,\ (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0}{} \ \mathsf{R}\mu \\
9: & \dfrac{X_1^0,\ X_2^0}{} \ \mathsf{Unfold}_\mu \\
10: & \dfrac{((P \wedge [a]Y) \vee \langle a\rangle X_1)^0,\ X_2^0}{} \ \mathsf{Unfold}_\mu \\
11: & \dfrac{((P \wedge [a]Y) \vee \langle a\rangle X_1)^0,\ ((\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0}{} \ \mathsf{R}\vee \\
12: & \dfrac{(P \wedge [a]Y)^0,\ (\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0}{} \ \mathsf{R}\vee \\
13: & \dfrac{(P \wedge [a]Y)^0,\ \langle a\rangle X_2^0}{} \ \mathsf{R}\wedge \\
14: & \dfrac{P^0,\ [a]Y^0,\ \langle a\rangle X_2^0}{\qquad} \ \mathsf{R}\langle\rangle \\
15: & \dfrac{Y^0,\ X_2^0}{} \ \mathsf{Unfold}_\nu \\
16: & \dfrac{\langle a\rangle Z \vee ((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0,\ X_2^0}{} \ \mathsf{R}\vee \\
17: & \dfrac{((\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1) \wedge (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2))^0,\ X_2^0}{} \ \mathsf{R}\wedge \\
18: & \dfrac{(\mu X_1.(P \wedge [a]Y) \vee \langle a\rangle X_1)^0,\ (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0,\ X_2^0}{} \ \mathsf{R}\mu \\
19: & \dfrac{X_1^0,\ (\mu X_2.(\neg P \wedge [a]Y) \vee \langle a\rangle X_2)^0,\ X_2^0}{} \ \mathsf{R}\mu \\
20: & \dfrac{X_1^0,\ X_2^0}{} \\
 & \text{UNSUCCESSFUL}
\end{array}
$$

Figure 5.8: An unsuccessful NUMU-tableau for Example 5.32.

Finiteness. Since there are finitely many possible goals in a tableau for ϕ, every tableau for ϕ must be finite.

Lemma 5.33. *Every NUMU-tableau for ϕ is a finite tree of degree $O(|\phi|)$ and height $2^{O(|\phi|)}$.*

Proof. Clearly, every tableau for ϕ is a finite tree whose degree is bounded by the number of $\langle\cdot\rangle$-subformulae of ϕ. There are no greater than $2^{2\cdot|\phi|}$ distinct goals in a tableau for ϕ. Hence any branch in a tableau for ϕ which is longer than $2^{2\cdot|\phi|}$ must contain a *proper* prefix $u_0, ..., u_i, ..., u_j$, $(i < j)$, such that u_i and u_j have the same goal. This means that u_j is a terminal. Thus every tableau for ϕ is of height $\leq 2^{2\cdot|\phi|} = 2^{O(|\phi|)}$. □

Soundness. The soundness of NUMU can be shown in the same way as in the tableau systems previously studied. Suppose $\mathcal{T}$ is a successful NUMU-tableau for a closed and guarded formula ϕ. The model $\mathcal{M}_\mathcal{T}$ (whose states are the modal nodes of $\mathcal{T}$) can be given as in Definition 4.14. We then prove that if $\mathcal{T}$ is a successful tableau, then $\mathcal{M}_\mathcal{T}$ is a model for ϕ. As before, this can be shown by proving that there is no μ-trail in $\mathcal{T}$. The notion of trails on a NUMU-tableau is very similar to Definition 4.15. In fact, we only need to replace the condition for rule Reset in Definition 4.15(b) by the following condition for rule Shift:

- For each node u where rule Shift is applied, if the formulae $\psi_1^0, ..., \psi_n^0$ in u are reduced to $\psi_1^1, ..., \psi_n^1$ in the child u', respectively, then $(u, \psi_i^0) \to (u', \psi_i^1)$ for each $i \leq n$.

Note that, as specified in Definition 4.15(f), for each terminal u with a companion v, we have $(u, \psi^\rho) \to (v, \psi^\rho)$ for each ψ^ρ in u.

Let us first make some observations concerning trails in $\mathcal{T}$. As before, let us call a trail in $\mathcal{T}$ in which a μ-variable is active an *active trail*.

Lemma 5.34. *Suppose there is an active trail in tableau* $\mathcal{T}$

$$(u_0, \psi_0^0) \to (u_1, \psi_1^{\rho_1}) \to ... \to (u_n, \psi_n^{\rho_n})$$

Then $\rho_1 = ... = \rho_n = 0$.

Proof. Suppose for some $i \geq 1$, $\rho_i = 1$ and $\rho_j = 0$ for all $j < i$. This means that rule Unfold_ν is applied at u_{i-1}, ψ_{i-1} is some ν-variable X, ψ_i is the unfolding of X, and no μ-variable is active in ψ_i. This contradicts the assumption that a μ-variable is active in the trail. □

Lemma 5.35. *Suppose u is a successful terminal with companion v, i.e.*

$$\begin{array}{l} v : \psi_1^0, ..., \psi_n^0, \gamma_1^1, ..., \gamma_m^1 \\ \qquad\vdots \qquad\qquad\qquad (\mathit{Shift}) \\ u : \psi_1^0, ..., \psi_n^0, \gamma_1^1, ..., \gamma_m^1 \end{array}$$

where rule Shift is applied between v and u. Then there can be no active trail from (v, ψ_i^0) to (u, ψ_j^0), for any $i, j \leq n$.

Proof. Obviously, if there is a trail from (v, ψ_i^0) to (u, ψ_j^0) in which a μ-variable Z is active, by the previous lemma, every goal between v and u must contain a formula of the form ψ^0 where Z is active in ψ. This would mean that Shift is not applicable at any node between v and u. □

Lemma 5.36. *Every successful NUMU-tableau for ϕ does* not *contain a μ-trail.*

Proof. Suppose $\mathcal{T}$ is a successful tableau for ϕ. Assume otherwise that there is a trail $(u_0, \phi_0^{\rho_0}) \to (u_1, \psi_1^{\rho_1}) \to (u_2, \psi_2^{\rho_2}) \to ...$ in which some μ-variable is active. Since $\mathcal{T}$ is successful (and finite), rule Shift must be applied at some node u_n, and hence $\rho_{n+1} = 0$. Therefore, by Lemma 5.34, $\rho_i = 0$ for all $i > n$. Since the tableau is finite, the suffix $(u_{n+1}, \psi_{n+1}^0) \to (u_{n+2}, \psi_{n+2}^0) \to ...$ of the trail must go through some path $v, ..., u$ where u is a successful terminal and v is its companion. But this contradicts the previous lemma. Hence $\mathcal{T}$ does not contain a μ-trail. □

Lemma 5.37. *If $\mathcal{T}$ does not contain μ-trail, then $\mathcal{M}_\mathcal{T}$ is a model of ϕ*

Proof. This can be shown in the same way as Lemma 4.46. □

Theorem 5.38 (Soundness of NUMU)**.** *Every closed and guarded formula which has a successful NUMU-tableau has a model in which the number of states is linear in the number of nodes in the tableau.*

Proof. Suppose $\mathcal{T}$ is a successful NUMU-tableau for a closed and guarded formula ϕ. By Lemma 5.36, there is no μ-trail in $\mathcal{T}$. By Lemma 5.37, $\mathcal{M}_\mathcal{T}$ is a model for ϕ. The model $\mathcal{M}_\mathcal{T}$ contains the modal nodes of $\mathcal{T}$ as its states; hence the size of $\mathcal{M}_\mathcal{T}$ is clearly linear in the number of nodes in $\mathcal{T}$. □

Completeness. The completeness proof for NUMU follows the same pattern as the proof of the small model theorem in the previous section. The outline of the proof is as follows. Suppose ϕ is a closed and guarded formula in Π_2^μ. First, we show that for any satisfiable set $\Phi = \{\phi_1, ..., \phi_n\}$ of subformulae of ϕ, there is a small tableau $\mathcal{T}_\Phi$ whose initial goal is $\phi_1^0, ..., \phi_n^0$ and all whose leaves are labelled with only formulae augmented by 1. By Lemma 5.34, this latter condition implies that there is no trail in which a μ-variable is active from the root of $\mathcal{T}_\Phi$ to any of its leaves. Clearly, this is similar to the property $(\star)$ of the finite subtrees used to construct a small model for ϕ (see Remark 5.24). We then join up those small tableaux $\mathcal{T}_\Phi$, for any satisfiable set Φ of subformulae of ϕ, to form a successful tableau for ϕ.

Lemma 5.39. *For any* satisfiable *set $\Phi = \{\phi_1, ..., \phi_n\}$ of subformulae of ϕ, there exists a* finite *tableau $\mathcal{T}_\Phi$ whose initial goal is $\phi_1^0, ..., \phi_n^0$ and such that*

(a) each goal is satisfiable,
(b) the goal at each terminal is a (consistent) set of literals and $[\cdot]$-formulae,

(c) the goal at each non-terminal leaf is of the form $\psi_1^1, ..., \psi_m^1, \Gamma$ *(*$m \geq 0$*), where* Γ *is a set of literals.*

Proof. Suppose $\Phi = \{\phi_1, ..., \phi_n\}$ is a satisfiable set of subformulae of ϕ. Hence there is a well-founded tree pre-model $\mathcal{P} = \langle \mathcal{S}, \Delta, \rightarrow \rangle$ whose root s_0 is annotated by Φ. By Proposition 5.23, there is a finite partial subtree $\mathcal{S}_\Phi$ of $\mathcal{S}$ rooted at s_0 such that

($\star$) for each leaf t of $\mathcal{S}_\Phi$, there is no active trail in $\mathcal{P}$ from s_0 to t, unless t is a terminal state.

We use the states in $\mathcal{S}_\Phi$ to guide the construction of $\mathcal{T}_\Phi$. To do so, we augment a goal with a state in $\mathcal{S}_\Phi$. In particular, a goal in the tableaux we are constructing will be of the form

$$s \vdash \gamma_1^{\rho_1}, ..., \gamma_m^{\rho_m} \qquad (m \geq 0),$$

where s is a state in $\mathcal{S}_\Phi$, satisfying the requirements:

(1) $\gamma_1, ..., \gamma_m$ are in the annotation of s.
(2) for each $i \leq m$, if a μ-variable is active in γ_i and $\rho_i = 0$, then there is an active trail in $\mathcal{P}$ from s_0 to (s, γ_i).

We start with the smallest tableau $\mathcal{T}_0$ whose root is labelled by $s_0 \vdash \phi_1^0, ..., \phi_n^0$. Suppose $\mathcal{T}_0, ..., \mathcal{T}_i$ are constructed. For each leaf v labelled with $s \vdash \Gamma$ in $\mathcal{T}_i$, if Γ contains only literals and $[\cdot]$-formulae or is of the form specified in (c), we do not expand v further. Otherwise, $s \vdash \Gamma$ must be in one of the forms below. Pick one applicable case and perform the described action.

- $s \vdash \psi^0, \psi^1, \Gamma$. Apply rule Thin to create a subgoal $s \vdash \psi^0, \Gamma$.
- $s \vdash (\psi_1 \wedge \psi_2)^\rho, \Gamma$. Apply rule $\mathsf{R}\wedge$ to create a subgoal $s \vdash \psi_1^\rho, \psi_2^\rho, \Gamma$.
- $s \vdash (\psi_1 \vee \psi_2)^\rho, \Gamma$. There is an $i \in \{1, 2\}$ such that $(s, \psi_1 \vee \psi_2) \rightarrow (s, \psi_i)$ in $\mathcal{P}$. Apply $\mathsf{R}\vee$ to create a subgoal $s \vdash \psi_i^\rho, \Gamma$.
- $s \vdash (\sigma X.\psi)^\rho, \Gamma$. Apply rule $\mathsf{R}\sigma$ to create subgoal $s \vdash X^\rho, \Gamma$.
- $s \vdash Z^\rho, \Gamma$, where Z identifies $\mu Z.\psi$. Apply rule Unfold_μ to create subgoal $s \vdash \psi^\rho, \Gamma$.
- $s \vdash X^\rho, \Gamma$, where X identifies $\nu X.\psi$. Apply rule Unfold_ν to create subgoal $s \vdash \psi^{\rho'}, \Gamma$ (where $\rho' = 1$ if there is no μ-variable active in ψ, otherwise $\rho' = \rho$).
- $s \vdash (\langle a_1 \rangle \psi_1)^{\rho_1}, ..., (\langle a_n \rangle \psi_n)^{\rho_n}, \Gamma$ (where $n > 1$ and Γ is a set of literals and $[\cdot]$-formulae). Since the goal is not of the form specified in (c), there must be a formula ψ^0 in the goal. By (2), there is an active trail from s_0 to (s, ψ). It follows from ($\star$) that s is not a leaf in $\mathcal{S}_\Phi$. Hence, for each $i \leq n$, $\mathcal{S}_\Phi$ must contain a state $t_i \in R_{a_i}(s)$ whose annotation contains $\{\psi_i\} \cup \{\psi \mid ([a_i]\psi)^\rho \in \Gamma\}$. Apply rule $\mathsf{R}\langle\rangle$ to create n subgoals $t_i \vdash \psi_i^{\rho_i}, \Gamma_{a_i}$ $(1 \leq i \leq n)$.

In all the above cases, it is clear that each expanded subgoal satisfies condition (1) and (2). Since ϕ is guarded and there are finitely many states in $\mathcal{S}_\Phi$, the construction must

terminate at some tableau $\mathcal{T}_m$. Let $\mathcal{T}$ be the tableau obtained from $\mathcal{T}_m$ by omitting the augmented states (i.e. replacing each goal $s \vdash \Gamma$ by Γ). $\mathcal{T}$ might contain a pair of nodes u, v along a branch having the same goal. We eliminate such pair by replacing the subtree rooted at u by the one at the lower node v. Repeat this process until no such pair of nodes u, v exists. We thus obtain a NUMU-tableau $\mathcal{T}_\Phi$ as required. □

Theorem 5.40 (Completeness of NUMU)**.** *Every satisfiable, closed, and guarded formula in* Π_2^μ *has a successful NUMU-tableau.*

Proof. Suppose ϕ is a satisfiable, closed, and guarded Π_2^μ-formula. Since ϕ is satisfiable, by the previous lemma, there is a finite tableau $\mathcal{T}_0$ whose initial goal is ϕ^0 and such that

(1) each goal is satisfiable,
(2) each terminal is successful,
(3) the goal at each non-terminal leaf is of the form $\phi_1^1, ..., \phi_n^1, \Gamma$ $(n \geq 1)$, where Γ is a set of literals.

We subsequently expand $\mathcal{T}_0$ while ensuring that (1) - (3) are satisfied. Suppose $\mathcal{T}_0, ..., \mathcal{T}_i$ are constructed. If all the leaves in $\mathcal{T}_i$ are terminals then, by (2), $\mathcal{T}_i$ is clearly a successful tableau for ϕ. Otherwise, we expand each non-terminal leaf u in $\mathcal{T}_i$ as follows. First, apply rule Shift to u to create a child u' whose goal is of the form $\phi_1^0, ..., \phi_n^0$. By (2), the set $\Phi = \{\phi_1, ..., \phi_n\}$ is satisfiable. Hence, by the previous lemma, there exists a finite tableau $\mathcal{T}_\Phi$ satisfying conditions (a) - (c) in the lemma. We replace u' by the tree $\mathcal{T}_\Phi$. If a node v in $\mathcal{T}_\Phi$ has the same goal as some ancestor v' in $\mathcal{T}_i$, we remove all the descendants of v. Clearly, v is now a successful terminal because rule Shift is applied between v' and v. Repeat this to each non-terminal leaf in $\mathcal{T}_i$. Since it has been shown that every NUMU-tableau is finite, the construction must terminate at some tableau $\mathcal{T}$, which by condition (2), is a successful tableau for ϕ. □

Chapter 6

Conclusion

In essence, the aim of this thesis is to find a decision procedure for the satisfiability problem and a proof of the small model theorem for the modal μ-calculus which do not appeal to automata-theoretic results. To certain extent, we have achieved our goal. However, there is still much room for improvement and further research.

Perhaps the main contribution of this thesis is the tableau system TS described in Chapter 4. The nicest feature of this tableau system is the fact that every tableau for any given formula is a finite tree structure. As a result, we are able to derive both the small model property and a decision procedure for satisfiability. We believe that other logical properties of the modal μ-calculus can also be obtained from TS. Of particular interest is how to prove the completeness of the axiom system $\mathbf{AX}$ from TS. This turns out to be more involved that expected. As explained (see Remark 4.65), the trick used in proving the axiomatic completeness for the aconjunctive fragment using tableau system ACON' fails to generalise to the full logic. The problem seems to be that the tableau system needs to be more intentional, i.e. recording more information of trail history into each goal. So far, it is still unclear how to refine the tableau system so that the axiomatic completeness can be shown. It is also interesting to see if TS is useful for proving other properties of the modal μ-calculus, such as the expressive-completeness result in [JW96] and the Craig interpolation property [dAH00].

The tableau system TS itself can still be improved further. Instead of having a set of names for each μ-variable, it should be sufficient for the μ-variables of the same *alternation depth* to share the names. This would slightly improve the bound on the small model to match the one obtained by automata-theoretic method (i.e. every satisfiable formula ϕ of alternation depth d is satisfied by a model with $2^{O(d|\phi|\log(|\phi|))}$ states). Also the nondeterministic algorithm which determines whether a formula has a successful tableau is far from being optimal. We believe that a deterministic exponential-time algorithm for this task exists. This would give the running time which matches the one obtained by the automata-theoretic method.

About the model-surgery techniques in Chapter 5, it seems that the notion of trail equivalence and safe pairs alone are not sufficient to prove the small model property.

As explained, we are only able to prove the result for linear models. It is interesting to learn if it is possible to define a relation on the states in a pre-model in the same way the goals in a tableau in TS are distinguished.

In addition, it is interesting to see if the techniques in this thesis can be applied for other variants of the modal μ-calculus, such as the modal μ-calculus with the past operators.

Bibliography

[AKM95] S. Ambler, M. Kwiatkowska, and N. Measor. *Duality and the completness of the modal μ-calculus*. Theoretical Computer Science 151, 3-27. 1995.

[AN01] A. Arnold and D. Niwinski. *Rudiments of μ-Calculus*. Studies in logic and the foundations of mathematics 146. Elsevier, 2001.

[BC96] G. Bhat and R. Cleaveland. *Efficient modal checking via the equational mu-calculus*. Proceedings of the Symposium on Logic in Computer Science (LICS'96), 304-312. IEEE 1996.

[BK95] M. Bonsangue and M. Kwiatkowska. *Re-interpreting the modal μ-calculus*. In A. Ponse, M. van Rijke, and Y. Venema (Eds.), *Modal Logic and Process Algebra*, 65-83. CSLI Lecture Notes, 1995.

[vBe76] J. van Benthem. *Modal Correspondence Theory*. PhD Thesis. University of Amsterdam, 1976.

[BdV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge tracts in theoretical computer science 53. Cambridge University Press, 2001.

[Bir93] G. Birkhoff. *Lattice Theory (third edition)*. American Mathematical Society, 1993.

[BS92] J. C. Bradfield and C. P. Stirling. *Local model checking for infinite state spaces*. Theoretical Computer Science 96, 157-174. 1992.

[Bra97] J. C. Bradfield. *The modal mu-calculus alternation hierarchy is strict*. Theoretical Computer Science 195, 133-153. 1997.

[Bra98] J. C. Bradfield. *Simplifying the modal mu-calculus alternation hierarchy*. Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS'98), LNCS 1373, 39-49. 1998.

[BS07] J. Bradfield and C. Stirling. *Modal Mu-Calculi*. Handbook of Modal Logic, 721-756. Elsevier, 2007.

[CE81] E. Clarke and E. Emerson. *Design and synthesis of synchronization skeletons using branching time temporal logic*. LNCS 131, 52-71. 1981.

[CES86] E. A. Clarke, E. A. Emerson, and A. P. Sistla. *Automatic verification of finite-state concurrent systems using temporal logic specifications*. ACM Transactions on Programming Languages and Systems. 8(2), 244-263. 1986.

[Che80] B. F. Chellas. *Modal logic: an introduction*. Cambridge University Press, 1980.

[Cle90] R. Cleaveland. *Tableau-based model checking in the propositional mu-calculus*. Acta Informatica 27, 725-747. 1990.

[dAH00] G. d'Agostino and M. Hollenberg. *Logical questions concerning the μ-calculus: interpolation, Lyndon and Łoś-Tarski*. Journal of Symbolic Logic 65(1), 310-332. 2000.

[Dam94] M. Dam. *CTL* and ECTL* as fragments of the modal μ-calculus*. Theoretical Computer Science 126, 77-96. 1994.

[DP90] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[EH86] E. Emerson and J. Halpern. *"Sometimes" and "not never" revisited: on branching time versus linear time*. Journal of the ACM 33, 151-178. 1986.

[EJ88] E. A. Emerson and C. S. Jutla. *The complexity of tree automata and logics of programs*. Proceedings of the 29th Symposium on the Foundation of Computer Science (FOCS '88), 328-337. IEEE, 1988.

[EL86] E. A. Emerson and C. Lei. *Efficient model checking in fragments of the propositional μ-calculus*. Proceedings of the 1st Symposium on Logic in Computer Science (LICS'86), 267-278. IEEE, 1986.

[Eme90] E. A. Emerson. *Temporal and modal logic*. In J. van Leeuwen (editor). *Handbook of Theoretical Computer Science: Volume B*, 995-1072. Elsevier, 1990.

[Fit83] M. Fitting. *Proof methods for modal and intuitionistic logics*. Riedel, 1983.

[FL79] M. J. Fischer and R. E. Ladner. *Propositional dynamic logic of regular programs*. Journal of Computing and System Sciences 18, 194-211. 1979.

[Flo67] R. W. Floyd. *Assigning meanings to programs*. Proceedings of the 19th Symposium in Applied Mathematics, 19-31. AMS, 1967.

[Gol92] R. Goldblatt. *Logics of time and computation*. CSLI Lectures 7. Center for the study of language and information, Stanford University, 1992.

[HC68] G. E. Hughes and M. J. Cresswell. *An introduction to modal logic*. Methuen and Co., 1968.

[HKT00] D. Harel, D. Kozen and J. Tiuryn. *Dynamic logic*. MIT Press, 2000.

[HM80] M. Hennessy and R. Milner. *On observing nondeterminism and concurrency.* Proceedings of the International Colloquium on Automata, Languages and, Programming (ICALP'80), LNCS 85, 295-309. 1980.

[HM85] M. Hennessy and R. Milner. *Algebraic laws for nondetrminism and concurrency.* Journal of the ACM 32, 137-162. 1985.

[Hoa69] C. A. R. Hoare. *An axiomatic basis for computer programming.* Communications of the ACM 12, 576-580, 583. ACM, 1969.

[Jur98] M. Jurdiński. *Deciding the winner in parity games is in* UP∩coUP. Information Processing Letters 68(3), 119-124. 1998.

[JW96] D. Janin and I. Walukiewicz. *On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic.* Proceedings of the International Conference on Concurrency Theory (CONCUR'96), LNCS 1119, 263-277. 1996.

[Kai97] R. Kaivola. *Using Automata to Characterise Fixpoint Temporal Logics.* PhD Thesis. University of Edinburgh, 1997.

[Kön27] D. König. Über eine schlubweise aus dem endlichen ins unendliche. Acta Litt. Science Szeged 3, 121-130, 1927.

[Koz83] D. Kozen. *Results on the propositional mu-calculus.* Theoretical Computer Science 27, 333-354. 1983.

[Koz86] D. Kozen. *A finite model theorem for the propositional μ-calculus.* Studia Logica XLVII, 233-241. 1986.

[KP84] D. Kozen and R. Parikh. *A decision procedure for the propositional μ-calculus.* LNCS 164, 313-325. 1984.

[Kru54] J. Kruskal. *The theory of well-partially-ordered sets.* Doctoral Thesis. Princeton University, June 1954.

[Kru60] J. Kruskal. *Well-Quasi-Ordering, The Tree Theorem, and Vazsonyi's Conjecture.* Transactions of the American Mathematical Society 95(2), 210-225. 1960.

[Kru72] J. Kruskal. *The Theory of Well-Quasi-Ordering: A Frequently Discovered Concept.* Journal of Combinatorial Theory (A) 13, 297-305. 1972.

[KVW00] O. Kupferman, M. Y. Vardi, and P. Wolper. *An Automata-Theoretic Approach to Branching-Time Model Checking.* Journal of the ACM 47, 312-360. 2000.

[Lar90] K. Larsen. *Proof systems for satisfiability in Hennessy-Milner logic with recursion.* Theoretical Computer Science 72, 265-288. 1990.

[Lav76] R. Laver. *Well-quasi-orderings and sets of finite sequences*. Proceedings of Cambridge Philosophical Society 79, 1-10. 1976.

[Len96] G. Lenzi. *A hierarchy theorem for the mu-calculus*. Proceedings of the International Colloquium on Automata, Languages and, Programming (ICALP'96), LNCS 1099, 87-109. 1996.

[LS01] M. Lange and C. Stirling. *Focus Games for Satisfiability and Completeness of Temporal Logic*. Proceedings of the 16th Symposium on Logic in Computer Science (LICS'01). IEEE, 2001.

[Mat02] R. Mateescu. *Local Model-Checking of Modal Mu-Calculus on Acyclic Labeled Transition Systems*. INRIA Research report 4430. 2002.

[MH84] S. Miyano and T. Hayashi. *Alternating automata on ω-words*. Theoretical Computer Science 32, 321-330. 1984.

[Mil80] R. Milner. *A calculus of communicating systems*. LNCS 92. 1980.

[Mil89] R. Milner. *Communication and concurrency*. Prentice Hall, 1989.

[MS87] D. E. Muller and P. E. Schupp. *Alternating automata on infinite trees*. Theoretical Computer Science 54, 267-276. 1987.

[Niw86] D. Niwinski. *On fixed point clones*. Proceedings of the International Colloquium on Automata, Languages and, Programming (ICALP'86), LNCS 226, 464-473. Springer, 1986.

[Niw88] D. Niwinski. *Fixed points vs. Infinite generation*. Proceedings of the 3rd Symposium on Logic in Computer Science (LICS'88), 402-409. 1988.

[NW97] D. Niwiński and I. Walukiewicz. *Games for the μ-calculus*. Theoretical Computer Science 163, 99-116. 1997.

[Par69] D. Park. *Fixpoint induction and proofs of program properties*. Machine Intelligence 5, 59-427. Edinburgh University Press, 1969.

[Par81] D. Park. *Concurrency and automata on infinite sequences*. Proceedings of the 5th GI Conference, 167-183. Springer, 1981.

[Pnu77] A. Pnueli. *The temporal logic of programs*. Proceedings of the 18th Symposium on the Foundation of Computer Science (FOCS'77), 46-57. IEEE, 1977.

[Pra76] V. Pratt. *Semantical considerations of Floyd-Hoare logic*. Proceedings of the 16th Symposium on the Foundation of Computer Science (FOCS'76), 109-121. 1976.

[Pra81] V. Pratt. *A decidable mu-calculus*. Proceedings of the 22nd Symposium on the Foundation of Computer Science (FOCS'82), 421-427. IEEE, 1982.

[Rab69] M. O. Rabin. *Decidability of second-order theories and automata on infinite trees*. Transactions of the AMS 141, 1-35. 1969.

[Rab72] M. O. Rabin. *Automata on Infinite Objects and Church's Problem*. AMS, 1972.

[Ram28] F. P. Ramsey. *On a problem of formal logic*. Proceedings of the London Mathematical Society 30, 338-384. 1929.

[Saf88] S. Safra. *On the complexity of ω-automata*. Proceedings of the 29th IEEE Symposium on Foundations of Computer Science, 319-327. 1988.

[SE89] R. S. Streett and E. A. Emerson. *An automata theoretic decision procedure for the propositional mu-calculus*. Information and Computation 81, 249-264. 1989.

[Smu68] R. M. Smullyan. *First-order logic*. Springer, 1968.

[Sti87] C. P. Stirling. *Modal logics for communicating systems*. Theoretical Computer Science 49, 311-347. 1987.

[Sti92] C. Stirling. *Modal and temporal logics*. In S. Abramsky, D. Gabbay, and T. Maibaum. *Handbook of Logic in Computer Science*, 477-563. Clarendon Press, 1992.

[Sti00] C. Stirling. *Modal and Temporal Properties of Processes*. Springer, 2000.

[Str81] R. Streett. *Propositional dynamic logic of looping and converse*. Proceedings of the 13th ACM Symposium on Theory of Computing (STOC'81), 375-383. 1981.

[SW91] C. Stirling and D. Walker *Local model checking in the modal mu-calculus*. Theoretical Computer Science 89, 161-177. 1991.

[Tar55] A. Tarski. *A lattice-theoretical fixpoint theorem and its application*. Pacific Journal of Mathematics 5, 285-309. 1955.

[Tho90] W. Thomas. *Automata on Infinite Objects*. In J. van Leeuwen (editor). *Handbook of Theoretical Computer Science: Volume B*, 133-191. Elsevier, 1990.

[Var88] M. Y. Vardi. *A Temporal Fixpoint Calculus*. Proceedings of the 15th ACM Symposium on Principles of Programming Languages (POPL'88), 250-259. ACM, 1988.

[VW86] M. Y. Vardi and P. Wolper. *Automata-theoretic techniques for modal logic of programs*. Journal of Computing and System Sciences 32, 183-221. 1986.

[VW94] M. Y. Vardi and P. Wolper. *Reasoning about infinite computations*. Information and Computation 115, 1-37. 1994.

[Wal93] I. Walukiewicz. *A Complete Deductive System for the μ-Calculus*. PhD Thesis. Warsaw University, 1993.

[Wal95] I. Walukiewicz. *Notes on the propositional μ-calculus: completeness and related results*. BRICS Notes NS-95-1. University of Aarhus, 1995.

[Wal00] I. Walukiewicz. *Completeness of Kozen's axiomatisation of the propositional μ-calculus*. Information and Computation 157, 142-182. 2000.

[Wal01] I. Walukiewicz. *Automata and Logic*. Lecture note for the EFF Summer School. 2001.

[Win89] G. Winskel. *Model checking the modal nu-calculus*. Proceedings of the International Colloquium on Automata, Languages and, Programming (ICALP'89). 1989.